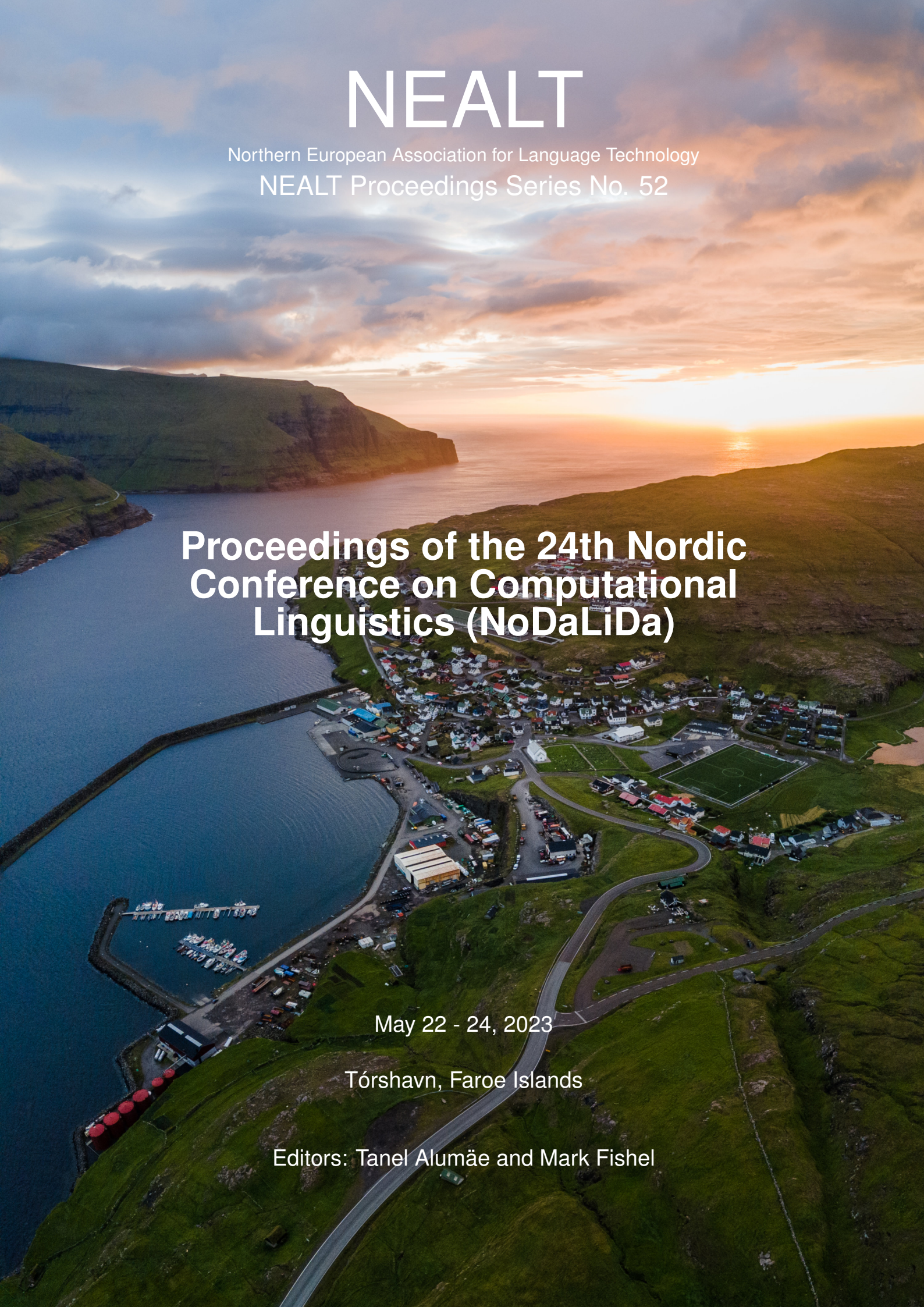


NEALT

Northern European Association for Language Technology

NEALT Proceedings Series No. 52

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)

May 22 - 24, 2023

Tórshavn, Faroe Islands

Editors: Tanel Alumäe and Mark Fishel

NoDaLiDa 2023

24th Nordic Conference on Computational Linguistics (NoDaLiDa)

Proceedings of the Conference

May 22-24, 2023

The NoDaLiDa organizers gratefully acknowledge the support from the following sponsors.

Silver

Bronze

Other

Front-cover photo: Luca Renner (@lucarennerphotography, http://visitfaroeislands.com)

Published by:

University of Tartu Library, Estonia
NEALT Proceedings Series, No. 52
Indexed in the ACL Anthology

ISBN: 978-99-1621-999-7
ISSN: 1736-8197 (Print)
ISSN: 1736-6305 (Online)

Volume Editors:
Tanel Alumäe and Mark Fishel

Message from the General Chair

It is my great pleasure and honor to welcome you to the 24th Nordic Conference on Computational Linguistics (NoDaLiDa 2023)!

After a couple of years' worth of conferences cancelled or held online (including the previous NoDaLiDa) we are extremely happy that NoDaLiDa 2023 is an onsite event. This is especially exciting given that for the first time in the history of NoDaLiDa conferences it takes place in Tórshavn, Faroe Islands.

The conference features three types of papers: long, short and demo papers. We are truly grateful to all the authors of papers submitted to this year's conference, with 130 papers submitted, a more than 40% increase over last year's yield! In total, we accepted 79 papers: 49 long papers, 26 short papers and 4 demo papers. More than half of the accepted papers are student papers, in which the first author is a student (29 long, 17 short and 2 demo papers). We would like to thank the 113 members of the program committee who reviewed the papers for their contributions!

The 79 accepted papers are grouped into 12 oral and 2 poster sessions. In addition to these regular sessions the conference program also includes three keynote talks. We would like to extend our gratitude to the keynote speakers for agreeing to present their work at NoDaLiDa. Georg Rehm from DFKI will talk on the topic of "Towards Digital Language Equality in Europe: An Overview of Recent Developments". Hjalmar P. Petersen will talk about "Aspects of the structure of Faroese". Marta R. Costa-Jussà from Meta will talk about "No-language-left-behind: Scaling Human-Centered Machine Translation and Toxicity at Scale".

The main conference program is preceded by three workshops: NLP for Computer-Assisted Language Learning (NLP4CALL), the Constraint Grammar Workshop and Resources and representations for under-resourced languages and domains (RESOURCEFUL'2023). We thank the workshop organizers for their efforts and for expanding the main conference program with a focus on more specific research topics.

I would like to express sincere gratitude to the entire team behind organizing NoDaLiDa 2023. I was honored to receive the invitation to serve as the general chair from the NEALT board; thank you for trusting me with this role. My deepest gratitude goes to Tanel Alumäe for serving as the publications chair and his active participation, Inguna Skadiņa for serving as the workshop chair as well as Iben Nyholm Debess for serving as the main local chair and smoothly handling all associated aspects of conference organization. I also want to thank the rest of the program chairs, Lilja Øvrelid and Christian Hardmeier and the local co-chairs Bergur Djurhuus Hansen, Peter Juel Henrichsen and Sandra Saxov Lamhauge. Thank you everyone for your contributions, you are awesome!

NoDaLiDa 2023 received financial support from several institutions and we would like to thank them here: NEALT, Dictus, Málráðið, Tórshavnar kommuna, BankNordik, Digitaliseringsstyrelsen, University of the Faroe Islands, Nationella språkbanken, Elektron and Formula.

Welcome and enjoy the 24th Nordic Conference on Computational Linguistics!

Mark Fishel, General Chair

Tartu

May 2023

Organizing Committee

General Chair

Mark Fishel, University of Tartu

Program Chairs

Tanel Alumäe, Tallinn University of Technology (Publication Chair)
Inguna Skadina, University of Latvia (Workshop Chair)
Christian Hardmeier, IT University of Copenhagen
Lilja Øvrelid, University of Oslo

Local Chair

Iben Nyholm Debess, University of the Faroe Islands

Local Co-Chairs

Bergur Djurhuus Hansen, University of the Faroe Islands
Peter Juel Henrichsen, Danish Language Council
Sandra Saxov Lamhauge, University of the Faroe Islands

Reviewers

Yvonne Adesam, Lars Ahrenberg, David Alfter, Krasimir Angelov, Ilze Auzina

Eduard Barbu, Jeremy Barnes, Valerio Basile, Ali Basirat, Timo Baumann, Jean-Philippe Bernardy, Arianna Bisazza, Jari Björne, Sidsel Boldsen, Gerlof Bouma, Gosse Bouma, Johan Boye, Chloé Braud, Maja Buljan

Marie Candito, Lin Chen, Mathias Creutz

Hercules Dalianis, Dana Dannélls, Leon Derczynski, Stefanie Dipper, Simon Dobnik

Adam Ek

Antske Fokkens

Filip Ginter, Rob Van Der Goot, Normunds Gruzitis, Tamás Grósz, Jon Gudnason

Mareike Hartmann, Daniel Hershcovich

Tommi Jauhiainen, Richard Johansson

Heiki-Jaan Kaalep, Kaarel Kaljurand, Katharina Kann, Jurgita Kapočiūtė-Dzikienė, Jussi Karlgren, Andre Kasen, Andreas Kirkedal, Roman Klinger, Mare Koit, Jiaming Kong, Marco Kuhlmann, Mikko Kurimo, Robin Kurtz, Andrey Kutuzov

Ekaterina Lapshinova-Koltunski, Krister Lindén, Pierre Lison, Hrafn Loftsson, Jan Tore Lønning

Bruno Martins, Farrokh Mehryary, Einar Meister, Hans Moen, Kadri Muischnek, Kaili Müürisep

Costanza Navarretta, Anna Björk Nikulasdottir, Joakim Nivre, Pierre Nugues, Arild Brandrud Næss

Heili Orav, Robert Östling

Patrizia Paggio, Anthi Papadopoulou, Eva Pettersson, Mārcis Pinnis, Barbara Plank, Sampo Pyysalo

Alessandro Raganato, Liisa Rätsep

Magnus Sahlgren, Askars Salimbajevs, Baiba Valkovska Saulīte, Yves Scherrer, Miikka Silfverberg, Kairit Sirts, Raivis Skadiņš, Maria Skeppstedt, Steinþór Steingrímsson, Sara Stymne, Torbjørn Svendsen, Rune Sætre

Jörg Tiedemann, Samia Touileb, Trond Trosterud, Andre Tättar

Martti Vainio, Daniel Varab, Martin Volk, Elena Volodina

Fredrik Wahlberg

Roman Yangarber

Niklas Zechner, Heike Zinsmeister

Invited Talk: Towards Digital Language Equality in Europe: An Overview of Recent Developments

Georg Rehm
German Research Center for Artifical Intelligence

Digital Language Equality (DLE) "is the state of affairs in which all languages have the technological support and situational context necessary for them to continue to exist and to prosper as living languages in the digital age", as we specified in one of our key reports of the EU project European Language Equality (ELE). Our empirical findings suggest that Europe is currently very far from having a situation in which all our languages are supported equally well through technologies. In this presentation, I'll give an overview of the two ELE projects and their main results and findings with a special focus on the Nordic languages (including insights from the FSTP projects supported through ELE2). This will also include a brief look back into the past, especially discussing the question if and where we have seen progress in the last, say, 15 years. Furthermore, I'll present an overview of our main strategic recommendations towards the European Union in terms of bringing about DLE in Europe by 2030. The presentation will conclude with a look at other relevant activities in Europe, including, critically the Common European Language Data Space project, which started in early 2023.

Invited Talk: No-language-left-behind: Scaling Human-Centered Machine Translation and Toxicity at Scale

Marta R. Costa-jussà
Meta AI

Machine Translation systems can produce different types of errors, some of which are characterized as critical or catastrophic due to the specific negative impact that they can have on users. In this talk, we focus on one type of critical error: added toxicity. We evaluate and analyze added toxicity in the context of NLLB-200 that open-sources models capable of delivering evaluated, high-quality translations directly between 200 languages. An automatic toxicity evaluation shows that added toxicity across languages varies from 0% to 5%. The output languages with the most added toxicity tend to be low-resource ones, and the demographic axes with the most added toxicity include sexual orientation, gender and sex, and ability. Making use of the input attributions allows us to further explain toxicity and our recommendations to reduce added toxicity are to curate training data to avoid mistranslations, mitigate hallucination and check unstable translations.

Invited Talk: Aspects of the Structure of Faroese

Hjalmar P. Petersen
University of Faroe Islands

Phonological changes and later morphologization have led to different complex alternations in Faroese. These are argued to emerge especially in small languages, with little contact and tight networks. The alternations will be exemplified with 'skerping', palatalization, glide insertion and the quantity-shift. There will be a discussion of the morphology-phonology interface, where the suggestion is that Faroese has 3 strata, stem1, stem2 and a word- strata. Syntactic variation and different construction will be addressed and illustrated; in this context reflexives are included and the present reorganization of the case system of complements of prepositions, where speakers use semantic and structural case in a certain way.

Table of Contents

Automated Claim Detection for Fact-checking: A Case Study using Norwegian Pre-trained Language Models

Ghazaal Sheikhi
MediaFutures
University of Bergen

Samia Touileb
MediaFutures
University of Bergen

Sohail Ahmed Khan
MediaFutures
University of Bergen

Abstract

We investigate to what extent pre-trained language models can be used for automated claim detection for fact-checking in a low resource setting. We explore this idea by fine-tuning four Norwegian pre-trained language models to perform the binary classification task of determining if a claim should be *discarded* or *upheld* to be further processed by human fact-checkers. We conduct a set of experiments to compare the performance of the language models, and provide a simple baseline model using SVM with tf-idf features. Since we are focusing on claim detection, the recall score for the *upheld* class is to be emphasized over other performance measures. Our experiments indicate that the language models are superior to the baseline system in terms of F1, while the baseline model results in the highest precision. However, the two Norwegian models, NorBERT2 and NB-BERT$_{\text{large}}$, give respectively superior F1 and recall values. We argue that large language models could be successfully employed to solve the automated claim detection problem. The choice of the model depends on the desired end-goal. Moreover, our error analysis shows that language models are generally less sensitive to the changes in claim length and source than the SVM model.

1 Introduction

With the growing concerns about misinformation, fact-checking has become an essential part of journalism. To mitigate the time and the human burden of fact-checking and to allow for more fact-checked articles, automated fact-checking (AFC) systems have been developed (Guo et al., 2022; Zeng et al., 2021; Lazarski et al., 2021). To approach automated fact-checking, three basic tasks are defined in the pipeline: claim detection, evidence retrieval, and claim verification. Claim detection refers to monitoring social media and political sources for identifying statements worth checking. The subsequent components retrieve reliable documents for debunking the detected claims and generate a verdict. Several tools have been developed to automate these tasks to meet the expectations of the human fact-checkers[1]. According to the studies on the user needs of fact-checkers, claim detection receives the highest preference among other AFC tools (Graves, 2018; Dierickx et al., 2022). Automated claim detection is a classification problem, where models are trained on sentences parsed from text documents and labelled by humans according to their check-worthiness (Hassan et al., 2017a).

In this work, we explore how well Norwegian pre-trained language models (LMs) perform on the task of automated claim detection. This is, to the best of our knowledge, the first attempt at automated claim detection for Norwegian using LMs. Fine-tuning LMs for the task of automated claim detection is not novel (Cheema et al., 2020; Zhuang et al., 2021; Shaar et al., 2021). However, this has never been done on Norwegian, and we believe that our insights into which errors these models do compared to simple baselines is a valuable contribution. Our research questions are:

- How well do Norwegian LMs perform on the task of automated claim detection compared to a simple SVM baseline?
- Which aspects of claim detection do these LMs still struggle with?

[1] https://www.rand.org/research/projects/truth-decay/fighting-disinformation/search.html

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 1–9
May 22-24, 2023

To address these questions, we first fine-tune each model on a small dataset from a Norwegian non-profit fact-checking organization, comprising claims manually annotated with labels reflecting their check-worthiness. Then we manually analyse the misclassifications of each model and provide an error analysis.

We believe that the contributions of this work have important societal implications. The case we study here sheds lights on the future directions of claim detection tools for fact-checking based on pre-trained language models for low to medium resourced languages. This would contribute to the fight against dis/misinformation by scaling and speeding up the fact-checking process.

The rest of the paper is organized as follows. In Section 2 we give an overview of previous work on automated claim detection. Section 3 describes the dataset and our experimental setup. We present and discuss our results and provide an error analysis in Section 4. Finally, we summarize our main findings, and discuss possible future works in Section 5.

2 Background

Automated claim detection for fact-checking does not have a long history, but it has turned to be one of the attractive fields of research in NLP (Hassan et al., 2015; Gencheva et al., 2017; Beltrán et al., 2021; Cheema et al., 2020; Shaar et al., 2021). One of the first studies on claim detection for AFC is initiated as part of the ClaimBuster project Hassan et al. (2017b). Their initial claim detection system was based on a set of features (sentiment, word count, part of speech (PoS) tags and named entities (NE)) followed by a feature selection and a traditional classifier namely Naive Bayes, SVM, and Random Forest(Hassan et al., 2015). Claim detection has also been addressed in languages other than English. ClaimRank is a claim detection system that supports both Arabic and English (Gencheva et al., 2017). A comprehensive set of features such as tf-idf, assertiveness, subjectivity, word embeddings are added to the ClaimBuster features and are fed to a two-layered neural network classifier (Gencheva et al., 2017).

In recent years, employment of pre-trained language models (LMs) in automated claim detection has been considered by numerous researchers (Cheema et al., 2020; Shaar et al., 2021; Beltrán et al., 2021). Several instances of these works are presented in the check-worthiness detection sub-tasks in CLEF CheckThat! editions (introduced in 2018 and ongoing) (Shaar et al., 2021; Nakov et al., 2022). CheckThat! provides data sets in different languages (English, Turkish, Arabic, Bulgarian, and Spanish) for the claim detection task on Twitter and political debates. The teams participating in this task have proposed classifier models mostly based on LMs. For instance, the top-ranked teams in CheckThat! 2020 used BERT (Devlin et al., 2019) and RoBERTa (Zhuang et al., 2021) with enhanced generalization capability (Williams et al., 2020) to detect check-worthy Tweets. For the task of detecting claims in political debates, the baseline BiLSTM (Schuster and Paliwal, 1997) model with GloVe embedding outperforms the LM-based systems (Martinez-Rico et al., 2020). ClaimHunter (Beltrán et al., 2021) is another BERT-based claim detection system that leverages XLM-RoBERTa [2](Conneau et al., 2020), a multilingual version of RoBERTa. It has been proved that the proposed model is superior to the classical baseline models NNLM+LR (Neural-Net Language Models embedding+Logistic Regression) and tf-idf+SVM.

To deal with the problem of small training data for LMs, data augmentation is employed. Claim detection from Twitter has been approached by generating synthetic check-worthy claims with lexical substitutions using BERT-based embeddings (Shaar et al., 2021). This approach improves the performance of BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) classification models (Shaar et al., 2021). It has also been shown that the BERTweet (Nguyen et al., 2020) model, fine-tuned on claims normalized and augmented by substitutions using WordNet, surpasses a reference n-gram model (Shaar et al., 2021).

3 Experiments

3.1 Data set

The data set is provided to us by Faktisk.no AS [3], a non-profit fact-checking organization and independent newsroom in Norway. Faktisk is jointly owned by several prominent Norwegian media houses, including VG, Dagbladet, NRK, TV2, Polaris Media, and Amedia. As per the company's articles of association, it operates under the overar-

[2] https://huggingface.co/xlm-roberta-large

[3] https://www.faktisk.no/om-oss

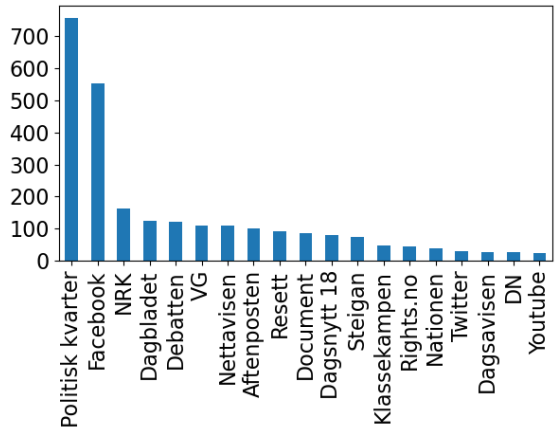


Figure 1: Most frequent sources of claims in our dataset provided by *Faktisk*, the non-profit fact-checking organization and independent newsroom in Norway.

ching ethical guidelines for the Norwegian press, as stipulated in the Vær Varsom poster [4]. To ensure its editorial and organizational independence, Faktisk.no adheres to the provisions of the Media Responsibility Act [5] and its articles of association. This ensures the editor's autonomy from the influence of the owners and other interested parties with interests in Faktisk's affairs. Thus, the funding news organizations of Faktisk and this project, being a source of some of the claims in the dataset should not raise concern about the independence of this research.

The data set comprises 4885 claims in Norwegian collected from social debates and public discourses from 04.03.2018 to 20.05.2022. Each claim in the dataset is provided with its respective source. These cover a selection of Norwegian newspapers (Dagbladet, VG, Nettavisen, Aftenposten, Klassekampen, Nationen, Dagsavisen, DN), alternative news outlets (Resett, Steigan, Document), think tank (Rights.no), the Norwegian Broadcasting Corporation (NRK), social media (Facebook, Twitter, YouTube), and TV/Radio (news) shows (Dagsnytt18, Politisk kvarter, Debatten). The alternative news outlets and the think tank are generally considered radical and controversial. The distribution of the occurrence of these sources can be seen in Figure 1.

A label is assigned to each claim, which refers to the actions taken by human fact-checkers. This data set has been labelled as part of the daily routine in the organization *Faktisk.no* and is neither hand-crafted nor crowd-sourced for training LMs. Thus, it resembles a real world problem. The data set labels are {*Discarded, Checked and rejected, Pre-checked, Published, Suspended, Checked, Facebook*}. After removing the missing values, the rare samples with label 'Facebook' (only nine claims), and the short claims with less than five words, we end up with 4116 claims across six different labels. These labels are produced during the fact-checking procedure. According to Faktisk, a claim must be based on verifiable information and should not be normative or a prediction of the future. For a claim to be considered for fact-checking, it must be supported by verifiable information and should not involve predictions or normative statements about the future. Additionally, the claim should have a certain degree of controversy and relevance to a majority of people. Less relevant claims may be fact-checked if they possess good entertainment value. Once a claim is selected, an attempt is made to contact the sender to verify the claim and its surrounding context. In cases where the sender is unknown, the origin and context of the claim are used as the starting point for the fact-checking process.

For our purposes, we aim to focus on class labels specified as whether a claim is worth being considered for further processing or if it should be discarded. We therefore define a binary classification task with the labels *Discarded* and *Upheld*; where the *Discarded* class refers to claims with the same label (Discarded) in the data set, and the *Upheld* class includes the claims originally labelled as *Pre-checked and rejected, Pre-checking, Published, Suspended, or Checking*. A brief explanation of these labels as well as the mapping of the original labels to the binary class labels is given in Table 1. The number of claims in each category is also presented. There are 2810 claims in the first class and 1306 claims in the second class. The average and the maximum length of claims in these samples are equal to 16 and 107 words, respectively.

3.2 Experimental setup

Pre-trained language models We fine-tune four Norwegian LMs to perform the binary classification task of claim detection. Norwegian

[4] http://presse.no/pfu/etiske-regler/vaer-varsom-plakaten/

[5] https://lovdata.no/dokument/NL/lov/2020-05-29-59

Class	Data Set Label	Description	#Claims
Discarded	Discarded	The claim has simply been discarded, there is no need for further investigation.	2810
Upheld	Pre-checked and rejected	Some preliminary work has been done to see if the claim is worth fact-checking, with a negative result.	372
	Pre-checking	Preliminary work to see if the claim is worth fact-checking has been started.	336
	Published	The fact-check about the claim has been published.	297
	Suspended	The claim will be taken up for consideration at a later time, and pre-checking or fact-checking will start then.	194
	Checking	A fact-check about the claim is in progress.	107

Table 1: Distribution of claims across class labels and related labels in our dataset.

has two official written standards: Bokmål and Nynorsk, and the four models are trained on data in both written forms. These are:

- **NorBERT** (Kutuzov et al., 2021): trained on the Norwegian newspaper corpus[6], and Norwegian Wikipedia, with a vocabulary of about two billion word tokens.
- **NorBERT2**[7]: trained on the non-copyrighted subset of the Norwegian Colossal Corpus (NCC)[8] and the Norwegian subset of the C4 web-crawled corpus (Xue et al., 2021). The size of the vocabulary is about 15 billion word tokens.
- **NB-BERT$_{\text{base}}$** (Kummervold et al., 2021): trained on the full NCC, and follows the architecture of the BERT cased multilingual model (Devlin et al., 2019). This model is bigger than the two previous ones, and comprises around 18.5 billion word tokens.
- **NB-BERT$_{\text{large}}$**[9]: trained on NCC, and follows the architecture of the BERT-large uncased model. This model is bigger and trained on more data (from the same sources) than it's base-form NB-BERT$_{\text{base}}$.

Training details The baseline model is a SVM classifier with tf-idf features (Jones, 2004), implemented using the Scikit-learn library[10]. To split the data, stratified sampling based on the original data set labels is employed to ensure the distributions of the real world label noise is consistent among the splits. The ratio of the train, validation, and test sets is $70\% - 20\% - 10\%$ respectively. The validation set is employed to tune the hyperparameters of the model. To account for class imbalance, weighted F1 is used for scoring, which computes metrics for individual labels and determine their weighted average based on their respective support values. The hyperparameters of the best model are `(C=100, gamma=0.1, kernel='rbf')`. It should be noted that the preliminary experiments revealed that the baseline model performs extremely poor on the minority class, *Upheld*. To make a fair comparison between the baseline model and the BERT-based models, we have examined five different random states for splitting the data and chosen the one in favour of the baseline model. Furthermore, we ensured that the distribution of the length of claims in the test split is consistent with the whole data set (See Figure 3 (a)). The same split is used for fine-tuning the pre-trained LMs. The selected split results in the highest F1 for the *Upheld* class by the baseline

[6] https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-4/

[7] https://huggingface.co/ltgoslo/norbert2

[8] https://github.com/NbAiLab/notram/blob/master/guides/corpus_description.md

[9] https://huggingface.co/NbAiLab/nb-bert-large

[10] https://scikit-learn.org/stable/

Hyperparameter	Value
batch_size	16
init_lr	2e-5
end_lr	0
warmup_proportion	0.1
num_epochs	5
max_seq_length	64

Table 2: Hyperparameter configuration of the four used Norwegian language models.

Model	t (s)	P	R	F1
tf-idf+SVM	2	**0.440**	0.168	0.243
NorBERT	44	0.328	0.626	0.430
NorBERT2	45	0.401	0.588	**0.477**
NB-BERT$_{\text{base}}$	48	0.358	0.336	0.345
NB-BERT$_{\text{large}}$	103	0.320	**0.740**	0.447

Table 3: Training time and claim detection results for the used models, in terms of precision (P), recall (R), and F1.

model among the five examined random splits.

The claim detection models are fine-tuned using a TensorFlow-based model for sequence classification [11] from the HuggingFace `transformers` library [12]. Bert-based model transformer have a sequence classification head, i.e. a linear layer on top. We use the same train, validation, and test splits as the baseline model and the validation set is deployed to return the best model after five epochs. All experiments are repeated for five times and the best run in terms of F1 is reported. All models are fine-tuned with Adam optimizer (Kingma and Ba, 2014). The other hyperparameter configurations are identical for all the four models, and can be seen in Tabel 2.

4 Results and discussion

4.1 Classification performance

The performance of the classification models on the test data are measured in terms of precision, recall, and F1. The *Upheld* class is treated as the positive class. It should be noted that in automated claim detection, overlooked important claims have a higher cost than misclassified unimportant claims. In other words, the recall score of the *Upheld* class should be given particular emphasis.

Table 3 presents the results for the baseline model and the four fine-tuned language models. Metrics are computed for the positive class. The highest score in each column is shown in bold. For the case of precision, the baseline system outperforms the LMs, but recall and F1 are extremely poor. It is noticeable how all the four LMs are superior to the baseline system in terms of F1, with NorBERT2 standing on the top. Another significant reflection of the results is NB-BERT$_{\text{large}}$'s superior performance in terms of recall. The training time (in seconds) is also given in the table. We run the experiments on a PC with an AMD Ryzen 7 5800X 8 Core Processor, an Nvidia GeForce RTX-3080 GPU with 10 GB graphics memory and 32 GB of RAM. The largest model, NB-BERT$_{\text{large}}$, requires twice as much training time compared to the other three models.

4.2 Error analysis

To get insights on the errors made by our models, confusion matrices of the predictions are plotted in Figure 2. The horizontal and vertical axes refer to the predicted and true labels, respectively. If we focus on one of the classes in terms of precision-recall, the baseline model and NB-BERT$_{\text{large}}$ are the best models. These models appear to learn one of the classes better, having fewer errors on that class. For example, NB-BERT$_{\text{large}}$ has learnt to correctly classify more instances of the upheld class. But the fact that it also classifies a large proportion of the claims from the discarded class as upheld shows that it simply has overfitted on the upheld class. This observation seems to be true for the SVM model (overfitted to the majority class) as well, and to some extent can be said about NB-BERT$_{\text{base}}$.

NorBERT and NorBERT2 seem to actually learn a more decent representation of the label distribution. While NorBERT exhibits some similarities with the previous models, by mostly classifying claims as one class rather than the other (in this case the discarded class), NorBERT2 seems to have a more balanced representation between the classes. It is the only model that is able to identify both classes to a certain degree, even if it still confuses many of the upheld claims as discarded claims. If we were to select a model that works

[11] `TFAutoModelForSequenceClassification`

[12] `https://huggingface.co/docs/transformers/index`

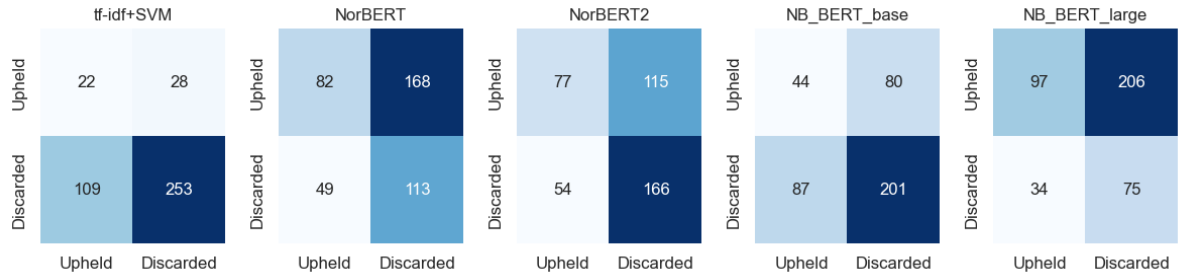
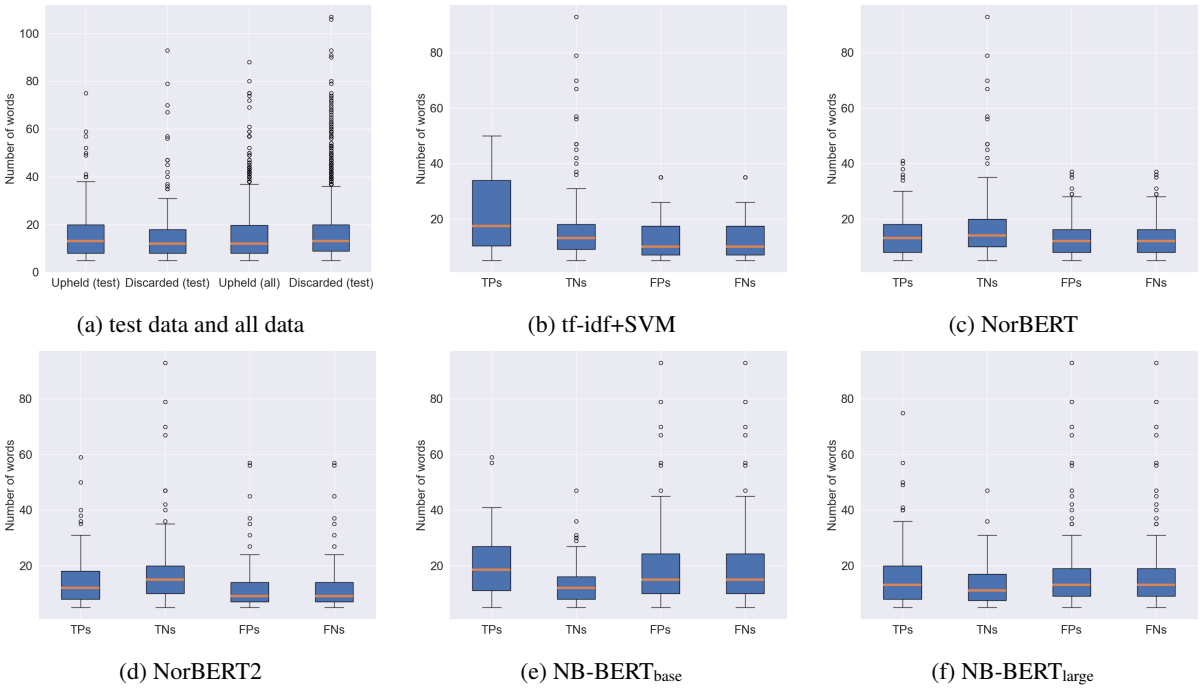


Figure 2: Confusion matrices of our models' predictions.



(a) test data and all data (b) tf-idf+SVM (c) NorBERT

(d) NorBERT2 (e) NB-BERT$_{base}$ (f) NB-BERT$_{large}$

Figure 3: Distribution of number of words in claims across true and false predictions for the four Norwegian language models and the SVM baseline.

fairly good on both classes, NorBERT2 would be the natural choice.

Further analysis is conducted on the length of the claims with respect to the model predictions for true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs). Figure 3 illustrates the box and whisker plots of the number of words in each of these groups. In Figure 3 (a), the number of words in the upheld and discarded class are shown for the test set and the whole data set. The length of the claims in the discarded class appears to be slightly larger than the upheld class. However, the quartiles and the median length are very close and thus length is not a significant discriminative feature. For the baseline model, length plays an important role in the model behaviour, though. The SVM model correctly classifies the longer claims from the upheld class and the shorter claims from the discarded class. Among the LMs, NorBERT and NB-BERT$_{large}$ are less sensitive to the length of the claims, as inferred from the similar statistics for true and false predictions. The figure also indicates that NorBERT2 suffers when predicting shorter claims, while NB-BERT$_{base}$ deteriorates for longer claims from the discarded class.

We also looked into the sources of the incorrectly classified claims for different models. The five most frequent sources in the data set, namely, 'Politisk kvarter', 'Facebook', 'NRK', 'Dagbladet', and 'Debatten' are considered. The percentage of the claims with false predictions from each source are shown in Figure 4. One interesting observation is claims from 'Facebook' are

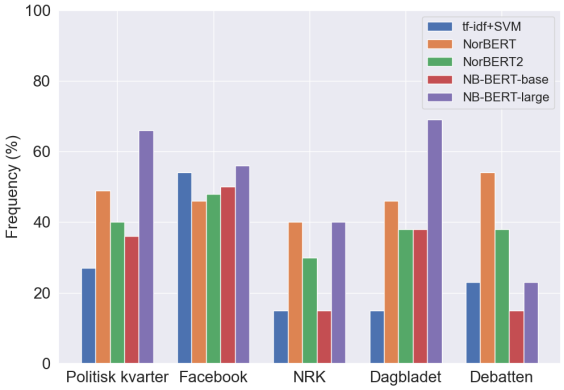


Figure 4: Percentage of the incorrectly classified claims from the five most frequent sources.

Data Set Label	#Claims	Acc.
Discarded	281	59.1%
Pre-checked and rejected	37	56.8%
Pre-checking	34	52.9%
Published	30	56.7%
Suspended	19	73.7%
Checking	11	45.5%

Table 4: Number of claims and accuracy in terms of original labels for the test set.

relatively difficult for all the models, while predicting the ones from 'NRK' seem to be more straightforward. This could be due to the differences in the writing styles in an official broadcasting organization and a social media platform. It is notable that the patterns for NorBERT and NorBERT2 are relatively similar across different sources. NB-BERT$_{\text{base}}$ and NB-BERT$_{\text{large}}$ are more sensitive to the source of the claims.

Finally, the predicted labels in the test set are analysed to see what percentage of each individual original label is correctly classified. We only focused on the NorBERT2 as it is the best model in terms of F1. Table 4 shows the number of claims in each category and the accuracy. The results are relatively comparable among the labels, which confirms the consistency of the mapping applied to convert the original labels to the binary labels. The two exceptions are Suspended and Checking class corresponding to the highest and the lowest accuracy, respectively.

5 Conclusion

In this work, we conduct a case study using Norwegian pre-trained LMs for the task of automated claim detection. Four existing Norwegian models in addition to an SVM baseline system are examined and compared using a claim detection data set that resembles a real world problem. The results show that language models outperform the baseline system. Different models can be selected for different purposes. If the overall performance is to be prioritized, the NorBERT2 model is the best performing. If the recall is the focus, then the biggest NB-BERT$_{\text{large}}$ model is to be selected.

Most of our observations can also be due to the differences between the LMs. The behaviour of our models can be due to model architecture, training procedures, and the datasets they were originally trained on. We also show how the length and the source of the claim plays a role in prediction patterns. We believe that there is more that can be uncovered from the behaviour of these models, and we plan to explore this in future works.

Limitations

Our work does have some limitations that might have impacted the outputs of our models. For instance, the behaviour of the models might partly be due to the skewed distribution of classes in the dataset, where the discarded class is the majority class. Another limitation is publishing the data to reproduce the results and perhaps to conduct further analysis. Faktisk provided the data set to us to investigate automated fact-checking systems and publish the results. At the moment, we are not permitted to make the data set publicly available, as it is part of the organization's internal procedure. This might hopefully change in the future.

Acknowledgements

We would like to express our deepest gratitude to Faktisk.no for sharing the data set with us for research purposes and for their insightful remarks on the fact-checking process.

This research was supported by industry partners and the Research Council of Norway with funding to MediaFutures: Research Centre for Responsible Media Technology and Innovation, through the Centres for Research-based Innovation scheme, project number 309339.

References

Javier Beltrán, Rubén Míguez, and Irene Larraz. 2021. Claimhunter: An unattended tool for automated claim detection on twitter. In *KnOD@WWW*.

Gullal S. Cheema, Sherzod Hakimov, and Ralph Ewerth. 2020. Check_square at checkthat! 2020 claim detection in social media via fusion of transformer and syntactic features. In *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020*, volume 2696 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Laurence Dierickx, Ghazaal Sheikhi, Duc Tien Dang Nguyen, and Carl-Gustav Lindén. 2022. Report on the user needs of fact-chekers. In *NORDIS Project Report: University of Bergen*, Task 4.2.

Pepa Gencheva, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cede no, and Ivan Koychev. 2017. A context-aware approach for detecting worth-checking claims in political debates. In *Proceedings of the 2017 International Conference on Recent Advances in Natural Language Processing*, RANLP '17, Varna, Bulgaria.

Lucas Graves. 2018. Understanding the promise and limits of automated fact-checking. *Reuters Institute for the Study of Journalism*.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017a. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1803–1812, New York, NY, USA. Association for Computing Machinery.

Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2015. Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 1835–1838, New York, NY, USA. Association for Computing Machinery.

Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, Vikas Sable, Chengkai Li, and Mark Tremayne. 2017b. Claimbuster: The first-ever end-to-end fact-checking system. *Proc. VLDB Endow.*, 10(12):1945–1948.

Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Per Egil Kummervold, Javier de la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. 2021. Operationalizing a national digital library: The case for a norwegian transformer model. In *Proc. of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa 2021)*.

Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-scale contextualised language modelling for norwegian. In *Proc. of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa 2021)*.

Eric Lazarski, Mahmood Al-Khassaweneh, and Cynthia Howard. 2021. Using nlp for fact checking: A survey. *Designs*, 5.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Juan R. Martinez-Rico, Lourdes Araujo, and Juan Martínez-Romo. 2020. Nlp&ir@uned at checkthat! 2020: A preliminary approach for check-worthiness and claim retrieval tasks using neural networks and graphs. In *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020*.

Preslav Nakov, Alberto Barrón-Cedeño, Giovanni Da San Martino, Firoj Alam, Julia Maria Struß, Thomas Mandl, Rubén Míguez, Tommaso Caselli, Mucahid Kutlu, Wajdi Zaghouani, Chengkai Li, Shaden Shaar, Gautam Kishore Shahi, Hamdy Mubarak, Alex Nikolov, Nikolay Babulkov, Yavuz Selim Kartal, Javier Beltrán, Michael Wiegand, Melanie Siegel, and Juliane Köhler. 2022. Overview of the CLEF-2022 CheckThat! lab on fighting the COVID-19 infodemic and fake news detection. In *Proceedings of the 13th International Conference of the CLEF Association: Information Access Evaluation meets Multilinguality, Multimodality, and Visualization*, CLEF '2022, Bologna, Italy.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Shaden Shaar, Maram Hasanain, Bayan Hamdan, Zien Sheikh Ali, Fatima Haouari, Mucahid Kutlu Alex Nikolov, Firoj Alam Yavuz Selim Kartal, Giovanni Da San Martino, Alberto Barrón-Cedeño, Rubén Míguez, Tamer Elsayed, and Preslav Nakov. 2021. Overview of the CLEF-2021 CheckThat! lab task 1 on check-worthiness estimation in tweets and political debates. In *Working Notes of CLEF 2021—Conference and Labs of the Evaluation Forum*, CLEF '2021, Bucharest, Romania (online).

Evan M. Williams, Paul Rodrigues, and Valerie Novak. 2020. Accenture at checkthat! 2020: If you say so: Post-hoc fact-checking of claims using transformer-based models. In *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020*, volume 2696 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online. Association for Computational Linguistics.

Xia Zeng, Amani S. Abumansour, and Arkaitz Zubiaga. 2021. Automated fact-checking: A survey. *Language and Linguistics Compass*, 15(10):e12438.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

Evaluating the Impact of Text De-Identification on Downstream NLP Tasks

Cedric Lothritz **Bertrand Lebichot** **Kevin Allix**
Saad Ezzini **Tegawendé F. Bissyandé** **Jacques Klein**
University of Luxembourg
6, rue Coudenhove-Kalergi
L-1359 Luxembourg
{cedric.lothritz,bertrand.lebichot,kevin.allix
saad.ezzini,tegawende.bissyande,jacques.klein} @uni.lu

Andrey Boytsov **Clément Lefebvre** **Anne Goujon**
BGL BNP Paribas
10, rue Edward Steichen
L-2540 Luxembourg
{andrey.boytsov,clement.c.lefebvre,anne.goujon } @bgl.lu

Abstract

Data anonymisation is often required to comply with regulations when transfering information across departments or entities. However, the risk is that this procedure can distort the data and jeopardise the models built on it. Intuitively, the process of training an NLP model on anonymised data may lower the performance of the resulting model when compared to a model trained on non-anonymised data. In this paper, we investigate the impact of de-identification on the performance of nine downstream NLP tasks. We focus on the de-identification and pseudonymisation of personal names and compare six different anonymisation strategies for two state-of-the-art pre-trained models. Based on these experiments, we formulate recommendations on how the de-identification should be performed to guarantee accurate NLP models. Our results reveal that de-identification does have a negative impact on the performance of NLP models, but it is relatively low. We also find that using pseudonymisation techniques involving random names leads to better performance across most tasks.

1 Introduction

Protection of personal data has been a hot topic for decades (Bélanger and Crossler, 2011). Careless sharing of data between companies, cyber-attacks, and other data breaches can lead to catastrophic leaks of confidential data, potentially resulting in the invasion of people's privacy and identity theft.

To mitigate damages and hold bad actors accountable, many countries introduced various laws that aim to protect confidential data, such as the Health Insurance Portability and Accountability Act (HIPAA) for healthcare confidentiality (Act, 1996), and the Gramm–Leach–Bliley Act (GLBA) in the financial domain (Cuaresma, 2002). Most notably, with the introduction of the General Data Protection Regulation (GDPR), the protection of personally identifiable information was codified into EU law. (Regulation, 2016) Failure to comply with these regulations can lead to huge fines in case of a data breach. Indeed, the amount of fines for GDPR violations adds up to over 1.5 trillion euros with the largest single fine of 746 million euros being imposed on Amazon.[1]

In order to mitigate data leaks, organisations such as financial institutes and hospitals are required to anonymise or pseudonymise sensitive data before processing them further. Similarly, automated NLP models should ideally be trained using anonymised data as resulting models could potentially violate a number of GDPR guidelines such as the individuals' right to be forgotten, and the right to explanation. Furthermore, models can be manipulated to partially recreate the training data (Song et al., 2017), which can result in disastrous data breaches. On the other hand, however, anonymisation of texts can lead to loss of information and meaning, making NLP models trained on anonymised data less reliable as a result (Meystre et al., 2014). Intuitively, this in turn could lead to a decrease in performance of such models when compared to models trained on non-anonymised

[1] at the time of writing this paper, according to https://www.privacyaffairs.com/gdpr-fines/

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 10–16

text. As such, it is crucial to choose an appropriate anonymisation strategy to lower this loss of information and avoid performance drops of models.

In this study, we investigate the impact of text de-identification on the performance of downstream NLP tasks, focusing on the anonymisation and pseudonymisation of person names only. This allows us to select from a wide array of NLP tasks as most datasets contain a large number of person names, whereas other types of names are less commonly found. Specifically, we compare six different anonymisation strategies, and two Transformer-based pre-trained model architectures in our experiments: the popular BERT (Devlin et al., 2018) architecture and the state-of-the-art ERNIE (Sun et al., 2020) architecture. Further, we look into nine different NLP tasks of varying degrees of difficulty.

We address the following research questions:

- RQ1: Which anonymisation strategy is the most appropriate for downstream NLP tasks?
- RQ2: Should a model be trained on original or de-identified data?

2 Experimental Setup

In this section, we present the datasets used in this study and we introduce the different anonymisation strategies that we compare against each other. We also show the pre-trained models we use.

2.1 Datasets

For this study, we selected several downstream tasks that greatly vary in complexity, ranging from simple text classification to complicated Natural Language Understanding (NLU) tasks featured in the GLUE benchmark collection (Wang et al., 2018). We ensured that each set contains a considerable number of person names. Most of these datasets are publicly available, except for a proprietary email classification dataset provided by our partners. Table 1 contains statistics about the datasets used for this study. We release the original as well as the de-identified datasets for most tasks.[2]

We choose three public classification tasks: Fake News Detection (FND)[3], News Bias Detection (NBD) (Bharadwaj et al., 2020), and Fraudulent Email Detection (FED) (Radev, 2008).

Five of our investigated tasks are featured in the GLUE collection, namely MRPC (Dolan and Brockett, 2005), RTE (Haim et al., 2006), WNLI (Levesque et al., 2012), CoLA (Warstadt et al., 2018), and MNLI (Williams et al., 2018).

Our final task is the Email Domain Classification Dataset (EDC) which we describe in greater detail. It is provided by our partners in the banking domain. As such, it is a proprietary dataset consisting of sensitive emails from clients, and thus cannot be publicly released. However, it serves as an authentic use-case for our study. The task consists of classifying emails along 19 broad domains related to banking activities such as *credit cards*, *wire transfers*, *account management* etc., which will then be forwarded to the appropriate department. We selected a subset of the provided dataset, such that each domain is represented equally. More specifically, for each domain in the set, we randomly selected $\simeq 500$ emails, for a total of nearly 9000 emails. Furthermore, the dataset is multilingual, but we perform our experiments on the emails written in French due to the high sample number.

2.2 Anonymisation Strategies

We consider six anonymisation strategies (AS1-6) for this study. These strategies are commonly found in the literature (Berg et al., 2020; Deleger et al., 2013). They largely fall into three categories: replacement by a generic token (AS1, AS2, AS3), removal of names (AS4), and replacement by a random name which we also refer to as pseudonymisation throughout this work (AS5, AS6). We describe each AS in Table2. Table 3 shows the differences between each AS on an example.

2.3 Name Detection

In order to detect names in the datasets, we fine-tune a *BERT Large* model on the task of Person Name Detection. We use the CoNLL-2003 dataset for Named Entity Recognition (Sang and De Meulder, 2003) and modify it by relabeling every non-*Person* entity as non-entity. The resulting training set consists of 204 567 words, 11 128 are *Person* entities and 193 439 are labeled as non-entities.[4] The resulting model achieved an F1 score of 0.9694, precision of 0.9786, and a recall of 0.9694 on the modified CoNLL-2003 test set. We use this fine-

[2] https://github.com/lothritz/anonymisation_paper

[3] https://www.kaggle.com/shubh0799/fake-news

[4] The dataset used to to train the de-identification model can be found at https://github.com/lothritz/anonymisation_paper/tree/main/anonymisation_model

dataset	FND	NBD	FED	MRPC	RTE	WNLI	CoLA	MNLI	EDC
train set	4382	1374	8980	3668	2489	635	6039	39 999	6354
dev set	690	196	997	407	276	71	851	5000	926
test set	1237	395	1926	1725	800	146	1661	5396	1798
#names	68 890	15 610	30 404	3324	3685	898	2600	85 999	6550
#unique	7500	3247	6104	1729	2042	102	335	10 460	2807
%de-identified	90.9	83.9	55.7	43.1	51	61.9	41	93.8	42.6
type	binary	multi	binary	binary	binary	binary	binary	multi	multi

Table 1: Statistics for the datasets. Size of datasets, number of names found in the training set (#names), number of unique names found in the training set (#unique), percentage of samples that contains at least one name (i.e. the percentage of samples to be de-identified) (%de-identified), and the type of the classification task (binary/multiclass)

Name	Description of AS
AS1	Singular generic token
AS2	Unique generic token for each name in document
AS3	Unique generic token for each distinct name in document
AS4	Removal of names
AS5	Random name for each name in document
AS6	Random name for each distinct name in document

Table 2: Description of Anonymisation strategies

tuned model to detect and replace names from the training, validation, and test set of the selected downstream tasks.

2.4 Model Training

We compare the impact of de-identification strategies using two Transformer-based models: BERT (Devlin et al., 2018) and ERNIE (Sun et al., 2020). For the tasks written in English, we use the uncased BERT Base mode and the ERNIE Base models. For the EDC task, we use the multilingual mBERT model and the ERNIE-M model published by Ouyang et al. (2021). For our study, we use the Transformers library by Huggingface (Wolf et al., 2019) as our framework. Furthermore, we take a grid-search based approach to determine the most appropriate fine-tuning parameters for each downstream task (cf. Appendix A)

3 Experimental Results

In this section, we show the results of our experiments and address the research questions from Section 1. For each task and for each pre-trained model, we fine-tune a model on the original dataset and each of our six anonymised datasets. We also de-identify the test sets accordingly and evaluate each model on the corresponding test set. We do five runs for each case, and average the results. We then compare the average performance for each AS to the performance of the models trained on original data. Table 4 shows the average performance of every model. For each of the GLUE tasks, we use the metric recommended by (Wang et al., 2018) and F1 score for the classification tasks.

3.1 Which anonymisation strategy is the most appropriate for downstream NLP tasks?

In order to determine the most appropriate strategy, we consider two ranking-based approaches: Borda Count and Instant Runoff (Taylor and Pacelli, 2008). For both approaches, we determine the score $s_{a,t}$ for each anonymisation strategy (AS, indexed by a) and for each task (indexed by t) in the following way: The best approach gets a score of five, the second best gets a score of four, etc.

The final *Borda Count* score for a given anonymisation strategy A is defined as $\sum_{t=0}^{T} s_{A,t}$ (where T is the total number of tasks, here, nine). The model with the highest total score is considered the best.

Instant Runoff is an iterative procedure. For each iteration, we count the number of wins for each AS, where an AS is considered a winner in a given task if its corresponding fine-tuned model outperforms every other model. We then eliminate the AS with the lowest number of wins and update the scores accordingly. We repeat this process until one AS remains, or until we cannot eliminate further ASs.

Table 5 shows the scores for each model and the winning anonymisation strategies according to the aforementioned approaches. For BERT models, we see that AS1, AS4, and AS6 are the best performing strategies according to Borda count, AS6 being a close winner. Instant Runoff leads to similar results with AS4 and AS6 reaching the final iteration, and AS6 being the overall winner. Furthermore, we note a lower variance in the scores for AS6

Original	"Hi, this is Paul, am I speaking to John?"	"Sorry, no, this is George. John is not here today."
AS1	"Hi, this is ENTNAME, am I speaking to ENTNAME?"	"Sorry, no, this is ENTNAME. ENTNAME is not here today."
AS2	"Hi, this is ENTNAME1, am I speaking to ENTNAME2?"	"Sorry, no, this is ENTNAME1. ENTNAME2 is not here today."
AS3	"Hi, this is ENTNAME1, am I speaking to ENTNAME2?"	"Sorry, no, this is ENTNAME3. ENTNAME2 is not here today."
AS4	"Hi, this is , am I speaking to "	"Sorry, no, this is . is not here today."
AS5	"Hi, this is Bert, am I speaking to Ernie?"	"Sorry, no, this is Elmo. Kermit is not here today."
AS6	"Hi, this is Jessie, am I speaking to James?"	"Sorry, no, this is Meowth. James is not here today."

Table 3: Example for each anonymisation strategy

		BERT							ERNIE						
Task	Metric	Original	AS1	AS2	AS3	AS4	AS5	AS6	Original	AS1	AS2	AS3	AS4	AS5	AS6
FND	F1	0.973	0.976↑	0.974↑	0.969↓	0.965↓	0.968↓	0.971↓	0.968	0.962↓	0.960↓	0.960↓	0.956↓	0.956↓	0.963↓
NBD	F1	0.653	0.658↑	0.647↓	0.654↑	0.681↑	0.674↑	0.683↑	0.678	0.681↑	0.684↑	0.695↑	0.709↑	0.653↓	0.669↓
FED	F1	0.994	0.995↑	0.996↑	0.996↑	0.996↑	0.994	0.995↑	0.996	0.994↓	0.993↓	0.994↓	0.993↓	0.995↓	0.993↓
MRPC	F1	0.791	0.786↓	0.769↓	0.768↓	0.797↑	0.792↑	0.783↓	0.811	0.824↑	0.817↑	0.799↓	0.832↑	0.826↑	0.820↑
RTE	Acc	0.691	0.670↓	0.654↓	0.639↓	0.624↓	0.644↓	0.666↓	0.703	0.696↓	0.665↓	0.671↓	0.683↓	0.716↑	0.676↓
WNLI	F1	0.520	0.530↑	0.526↑	0.551↑	0.586↑	0.541↑	0.535↑	0.561	0.472↓	0.557↓	0.564↑	0.595↑	0.614↑	0.550↓
CoLA	MCC	0.555	0.520↓	0.522↓	0.524↓	0.443↓	0.495↓	0.532↓	0.519	0.517↓	0.543↑	0.556↑	0.385↓	0.540↑	0.542↑
MNLI	Acc	0.754	0.742↓	0.730↓	0.734↓	0.745↓	0.742↓	0.747↓	0.789	0.774↓	0.750↓	0.759↓	0.770↓	0.776↓	0.773↓
EDC	F1	0.626	0.624↓	0.683↑	0.617↓	0.619↓	0.616↓	0.595↓	0.642	0.635↓	0.696↑	0.642	0.635↓	0.627↓	0.621↓

Table 4: Results of our fine-tuned models. We highlight in green (↑) the models that outperform the models trained on original data, in red (↓) the models that do not.

	BERT						ERNIE					
Task	AS1	AS2	AS3	AS4	AS5	**AS6**	AS1	AS2	AS3	AS4	**AS5**	AS6
FND	5	4	2	0	1	3	4	3	3	1	1	5
NBD	2	0	1	4	3	5	2	3	4	5	0	1
FED	2	5	5	5	0	2	4	2	4	2	5	2
MRPC	3	1	0	5	4	2	3	1	0	5	4	2
RTE	5	3	1	0	2	4	4	0	1	3	5	2
WNLI	1	0	4	5	3	2	0	2	3	4	5	1
CoLA	2	3	4	0	1	5	1	4	5	0	2	3
MNLI	3	0	1	4	3	5	4	0	1	2	5	3
EDC	4	5	2	3	1	0	3	5	4	3	1	0
Total	27	21	20	26	18	**28**	25	20	25	25	**28**	21
Avg.	3	2.33	2.22	2.89	2	**3.11**	2.78	2.22	2.78	2.78	**3.11**	2.33

Table 5: Ranking scores for fine-tuned models. **Bold text** shows the winner according to Borda Count, underlined text according to Instant Runoff.

		BERT							ERNIE						
Task	Metric	Original	AS1	AS2	AS3	AS4	AS5	AS6	Original	AS1	AS2	AS3	AS4	AS5	AS6
FND	F1	0.973	0.933↓	0.910↓	0.907↓	0.950↓	0.963↓	0.963↓	0.968	0.951↓	0.938↓	0.935↓	0.957↑	0.967↑	0.967↑
NBD	F1	0.653	0.566↓	0.551↓	0.546↓	0.601↓	0.602↓	0.609↓	0.678	0.683	0.684	0.659↓	0.687↓	0.683↑	0.683↑
FED	F1	0.994	0.995	0.995	0.995	0.996	0.996	0.996	0.996	0.995	0.995	0.995	0.996	0.996	0.996
MRPC	F1	0.791	0.809↑	0.811↑	0.811↑	0.819↑	0.816↑	0.814↑	0.811	0.848↑	0.848↑	0.849↑	0.852↑	0.804↓	0.834↑
RTE	Acc	0.691	0.665↓	0.663↑	0.669↑	0.670↑	0.645↑	0.660↓	0.700	0.703↑	0.701↑	0.693↑	0.699↑	0.688↓	0.704↑
WNLI	F1	0.520	0.504↓	0.504↓	0.504↓	0.504↓	0.504↓	0.504↓	0.561	0.435↓	0.442↓	0.467↓	0.506↓	0.458↓	0.428↓
CoLA	MCC	0.555	0.376↓	0.515↓	0.528↑	0.335↓	0.549↑	0.550↑	0.519	0.427↓	0.537↓	0.511↓	0.313↓	0.518↓	0.523↓
MNLI	Acc	0.754	0.753↑	0.724↓	0.753↑	0.753↑	0.744↑	0.744↓	0.789	0.783↑	0.545↓	0.760↑	0.772↑	0.669↓	0.765↓

Table 6: Results of testing the original models on de-identified data. We highlight in green (↑) the models that significantly outperform the matching model in Table 4 using a Wilcoxon test, in red (↓) the models that perform significantly worse, in black the models that do not perform significantly differently.

when compared to AS4. In contrast, when evaluating ERNIE models, we note that AS5 models are performing significantly better than every other strategy according to Borda Count. Similarly, AS5 also wins the Instant Runoff with AS4 and AS5 making it to the final round. Overall, it appears that using random names over generic tokens to de-identify textual data is the preferable solution as AS1, AS2, AS3 models, which were all trained on data with generic tokens, usually rank low.

3.2 Should a model be trained on original or de-identified data?

In order to answer this question, we investigate the performance of models trained on original data on the de-identified test sets (cf. Table 4) and compare them to the models trained directly on de-identified data. Table 6 shows the results of testing models

trained on original training sets and evaluated on each of the de-identified test sets. We find that nearly half of the models trained on de-identified data outperform the counterpart model trained on original data. While there is not always a clear trend, we observe that the original models almost consistently perform better in the MRPC and RTE tasks, and perform worse in the WNLI and CoLA tasks, regardless of the architecture used. Furthermore, for BERT models, the models trained on de-identified data consistently perform worse on the FND and NBD tasks. For the ERNIE models, the models trained on original data consistently perform better on the FED task ever so slightly. Despite these observations, we notice that the performance losses are oftentimes very high, specifically for the NBD, WNLI, and CoLA tasks, while performance gains tend to be lower.

4 Discussion

Judging by the results of our experiments, we recommend practitioners to de-identify their sensitive textual data using random names, as they typically lead to the best results among the anonymisation strategies we tested. We also recommend to de-identify data before the training of NLP models. It follows that it is important to keep the de-identification process and naming schemes consistent throughout the entire pipeline that uses the data in order to mitigate potential performance losses of models. It may also be important to keep the number of names sufficiently high in order to avoid introducing bias in the training that may contribute to unfair discrimination against specific names, a well-known issue in machine learning models that handle person names (Caliskan et al., 2017).

5 Related Work

Relevant studies done on textual data largely focus on medical texts and on a very limited number of tasks and anonymisation strategies when compared to our work. On the other hand, they typically anonymise a wide variety of protected health information (PHI) classes, while our work focuses on anonymisation of persons' names only. Berg et al. (2020) studied the impact of four anonymisation strategies (pseudonymisation, replacement by PHI class, masking, and removal) on downstream NER tasks for the clinical domain. Similarly to our findings, they find that pseudonymisation yields the best results among the investigated strategies. On the other hand, removal of names resulted in the highest negative impact on the downstream tasks. Deleger et al. (2013) investigated the impact of anonymisation on an information extraction task using a dataset of 3503 clinical notes. They anonymised 12 types of PHI such as patients' name, age, etc., and used two anonymisation strategies (replacement by fake PHI, and masking). They found no significant loss in performance for this task. Similarly, Meystre et al. (2014) found that the informativeness of medical notes only marginally decreased after anonymisation, using 18 types of PHI and 3 anonymisation strategies (replacement by fake PHI, replacement by PHI class, and replacement by *PHI* token). Using the same anonymisation strategies and ten types of PHI, Obeid et al. (2019) investigated the impact of anonymisation on a mental status classification task. Comparing nine different machine learning models, they did not find any significant difference in performance between original and anonymised data.

6 Conclusion

In this paper, we conducted an empirical study analysing the impact of de-identification on downstream NLP tasks. We investigated the difference in performance of six anonymisation strategies on nine NLP tasks ranging from simple classification tasks to hard NLU tasks. Further, we compared two architectures, BERT and ERNIE. Overall, we found that de-identifying data before training an NLP model does have a negative impact on its performance. However, this impact is relatively low. We determined that pseudonymisation techniques involving random names lead to higher performances across most tasks. Specifically, replacing names by random names (AS5) had the least negative impact when using an ERNIE model. Similarly, replacing by random names while preserving the link between identical names (AS6) worked best for BERT models. We also showed that it is advisable to de-identify data prior to training as we observed a large difference in performance between models trained on original data versus de-identified data. There is also a noticeable difference between the performances of BERT and ERNIE, warranting further investigation into the performance differences between a larger number of language models.

References

Accountability Act. 1996. Health Insurance Portability and Accountability Act of 1996. *Public law*, 104:191.

France Bélanger and Robert E Crossler. 2011. Privacy in the digital age: a review of information privacy research in information systems. *MIS quarterly*, pages 1017–1041.

Hanna Berg, Aron Henriksson, and Hercules Dalianis. 2020. The impact of de-identification on downstream named entity recognition in clinical text. In *11th International Workshop on Health Text Mining and Information Analysis*, pages 1–11. Association for Computational Linguistics.

Avinash Bharadwaj, Brinda Ashar, Parshva Barbhaya, Ruchir Bhatia, and Zaheed Shaikh. 2020. Source based fake news classification using machine learning.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Jolina C Cuaresma. 2002. The Gramm-Leach-Bliley Act. *Berkeley Tech. LJ*, 17:497.

Louise Deleger, Katalin Molnar, Guergana Savova, Fei Xia, Todd Lingren, Qi Li, Keith Marsolo, Anil Jegga, Megan Kaiser, Laura Stoutenborough, et al. 2013. Large-scale evaluation of automated clinical note de-identification and its impact on information extraction. *Journal of the American Medical Informatics Association*, 20(1):84–94.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional Transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Roy Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second PASCAL Recognising Textual Entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 7.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Stéphane M Meystre, Oscar Ferrández, F Jeffrey Friedlin, Brett R South, Shuying Shen, and Matthew H Samore. 2014. Text de-identification for privacy protection: a study of its impact on clinical text information content. *Journal of biomedical informatics*, 50:142–150.

Jihad S Obeid, Paul M Heider, Erin R Weeda, Andrew J Matuskowitz, Christine M Carr, Kevin Gagnon, Tami Crawford, and Stephane M Meystre. 2019. Impact of de-identification on clinical text classification using traditional and deep learning classifiers. *Studies in health technology and informatics*, 264:283.

Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE-M: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 27–38, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dragomir Radev. 2008. CLAIR collection of fraud email (repository) - ACL Wiki.

Protection Regulation. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council. *Regulation (eu)*, 679:2016.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. 2017. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security*, pages 587–601.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8968–8975.

Alan D Taylor and Allison M Pacelli. 2008. *Mathematics and politics: strategy, voting, power, and proof*. Springer Science & Business Media.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

7 Appendices

7.1 Appendix A: Fine-Tuning Hyperparameters

	BERT			ERNIE		
Task	batch size	learn rate	#epoch	batch size	learn rate	#epoch
FND	16	5e-5	1	8	2^{-5}	1
NBD	16	5e-5	3	8	2^{-5}	5
FED	32	3e-5	3	32	5^{-5}	1
MRPC	16	5e-5	3	32	3^{-5}	4
RTE	16	5e-5	4	4	2^{-5}	4
WNLI	16	3e-5	4	8	2^{-5}	4
ColA	16	5e-5	3	64	3^{-5}	3
MNLI	16	5e-5	2	512	3^{-5}	3
EDC	16	5e-5	5	8	3^{-5}	3

Table 7: Hyperparameters for fine-tuning pre-trained models for downstream tasks

Abstractive Text Summarization for Icelandic

Þór Sverrisson
Department of Computer Science
University of Iceland
Iceland
ths220@hi.is

Hafsteinn Einarsson
Department of Computer Science
University of Iceland
Iceland
hafsteinne@hi.is

Abstract

In this work, we studied methods for automatic abstractive summarization in a low-resource setting using Icelandic text, which is morphologically rich and has limited data compared to languages such as English. We collected and published the first publicly available abstractive summarization dataset for Icelandic and used it for training and evaluation of our models. We found that using multilingual pre-training in this setting led to improved performance, with the multilingual mT5 model consistently outperforming a similar model pre-trained from scratch on Icelandic text only. Additionally, we explored the use of machine translations for fine-tuning data augmentation and found that fine-tuning on the augmented data followed by fine-tuning on Icelandic data improved the results. This work highlights the importance of both high-quality training data and multilingual pre-training in achieving effective abstractive summarization in low-resource languages.

1 Introduction

The task of automatic text summarization has been gaining interest in recent years due to the increasing amount of available information and the need for well-written summaries that preserve key information while being coherent and flowing naturally. Two main approaches to automatic text summarization are extractive and abstractive methods. Extractive methods compose the summary out of copies of important sections from the original text, whereas abstractive methods rephrase and shorten the text similar to how a human would (Tas and Kiyani, 2017). The rise of Transformer models (Vaswani et al., 2017) in natural language processing (NLP) has led to great advances in the field, particularly in abstractive summarization (Zhang et al., 2020). However, these models often rely on a large amount of text data and computational resources for pre-training. This raises the question of whether low-resource languages can build advanced NLP models for summarization, given the lack of data.

We aim to address this question by studying the use of state-of-the-art Transformer models for abstractive summarization of Icelandic text. We introduce the first publicly available abstractive summarization dataset for Icelandic, RÚV Radio News (RRN), and use it for training and evaluation of the models. With that approach, we aim to study whether state-of-the-art Transformer models can be adapted to perform abstractive summarization in a low-resource setting for Icelandic text. In order to support future research on abstractive summarization in Icelandic, we are sharing our dataset[1] and the fine-tuned model[2] with the research community.

This work is motivated by the increasing demand for automatic text summarization and the challenges of applying machine learning methods to low-resource languages such as Icelandic. The study of NLP in low-resource languages is important for language preservation, and this research contributes to this field by providing a dataset for Icelandic and evaluating the performance of state-of-the-art Transformer models on it. Summarization has been claimed to be challenging in low-resource settings (Zoph et al., 2016; Khurana et al., 2022) and the potential solution that we base our work on is to apply transfer learning (Zhuang et al., 2021) and data augmentation techniques (Tanner and Wong, 1987).

[1] https://huggingface.co/datasets/thors/RRN

[2] https://huggingface.co/thors/mt5-base-icelandic-summarization

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 17–31

2 Background

Abstractive summarization is a complex task that involves identifying important information from a text and expressing it in new words. The Transformer architecture (Vaswani et al., 2017), which is based on the attention mechanism (Bahdanau et al., 2015), has become popular for this task as it can efficiently work with larger text segments and take into account context in the input.

Transformers are widely applied through transfer learning, a technique introduced by Yosinski et al. (2014) where a model trained on one task is fine-tuned or reused as the starting point for a model on a similar or different task. Prior to the transfer, the models are generally trained using self-supervision, which allows the models to leverage a large, diverse corpus of unlabeled text data. For generative models, the pre-training objective often involves masking parts of the input sequence and tasking the model with filling in the gaps, as proposed by (Song et al., 2019) for example. Raffel et al. (2020) demonstrated with the T5 model that many NLP problems can be treated as text-to-text tasks, allowing for the pre-training of a single encoder-decoder Transformer on a diverse set of tasks. Additionally, BART models (Lewis et al., 2020) have been trained to reconstruct a text document that has been corrupted with an arbitrary noising function and have proved to be very effective at tasks such as summarization. The PEGASUS model (Zhang et al., 2020) uses a pre-training objective that closely resembles the summarization task, resulting in a model that adapts faster when fine-tuned on a small number of examples.

Pre-training language models through self-supervised learning has achieved impressive results when applied to abstractive summarization tasks. However, obtaining high-quality summarization outcomes can be difficult when there is a scarcity of data for fine-tuning, a common issue encountered with low-resource languages. To tackle this challenge, researchers have turned to transfer learning and data augmentation techniques, which have proven to be effective in various low-resource natural language processing (NLP) tasks (Hedderich et al., 2021). Prior results on abstractive summarization in a low-resource setting serve as good examples of applying such methods (Fadaee et al., 2017; Sennrich et al., 2016).

Transfer learning methods have enabled progress in Icelandic NLP tasks, such as translation (Símonarson et al., 2021), question answering (Snæbjarnarson and Einarsson, 2022b), and named entity recognition (Snæbjarnarson et al., 2022). However, research on Icelandic summarization has predominantly concentrated on extractive approaches (Christiansen, 2014; Daðason et al., 2021; Daðason and Loftsson, 2022). Multilingual models, like XLM-R (Conneau et al., 2020) and mT5 (Xue et al., 2021), have exhibited promising results across a wide range of NLP tasks and have been particularly advantageous for Icelandic tasks (Snæbjarnarson et al., 2022; Snæbjarnarson and Einarsson, 2022a).

3 Methods

3.1 Data

A summary of the text corpora utilized in this study is provided in Table 1. The English language corpora were translated to Icelandic using machine translation, as described in Section 3.1.4.

3.1.1 Pre-training Corpus

The **Icelandic Gigaword Corpus** (IGC, (Steingrímsson et al., 2018)) version 20.05 was used for pre-training of the Gullfaxi model (see Section 3.2). The corpus consists of a collection of approximately 5 million documents from various categories, including adjudications, parliamentary speeches, news, books, and scientific journals. The corpus consists of text that is automatically divided into sentences and running words, tagged, and lemmatized. The IGC-News1 21.05 dataset, consisting of news articles from the year 2020, was used for validation during pre-training. These articles were not included in the training data.

3.1.2 Fine-tuning corpora

In this study, we utilized the following news summarization datasets for fine-tuning our models:

RÚV Radio News (RRN) dataset, which consists of news stories from the Icelandic National Broadcasting Service (RÚV) collected specifically for this study. It includes 4k stories from 2021 and 2022, containing many stories related to COVID-19 and domestic news.

XSum dataset (Narayan et al., 2018), which features a variety of English-language BBC articles from 2010 to 2017, each accompanied by a professional, single-sentence summary.

CNN/DailyMail dataset (Hermann et al., 2015), which includes English-language news stories

Dataset	# Documents	Language	Type
IGC 20.05	5M	is	Generic
IGC-News1 21.05 (2020)	112k	is	Generic
RRN	4k	is	Summarization
XSum	227k	en	Summarization
CNN/DailyMail	311k	en	Summarization

Table 1: Overview of the datasets used in this study. The language column refers to the original language of the dataset.

from CNN and Daily Mail websites, each accompanied by human-written summary bullets.

Note that there was no overlap between the fine-tuning datasets and the pre-training corpus. We study fine-tuning on the datasets separately and we also study fine-tuning on translated data followed by fine-tuning on RRN.

3.1.3 Pre-processing RRN

The Icelandic National Broadcasting Service (RÚV) granted access to a database of news stories via a custom interface that was available on-premises at their headquarters. The stories were manually selected from the database, and only transcripts of radio news from 2021 and 2022 were used. The RRN dataset was extracted from these transcripts and comprises four parts for each story: a title, an intro, the main story, and a summary. To ensure that the dataset was suitable for the summarization task, we filtered out stories that were not relevant, such as live broadcasts and weather news. Additionally, we programmatically removed reporters' comments, phone numbers, and instructions for the broadcast. The intro and the summary are often similar as they both provide an overview of the key points of the story and in some instances, they may be identical. For a given date, the summaries were in a separate document and not linked to a story by any unique identifier. Therefore, the summaries were paired with their corresponding stories in a heuristic manner using a ROUGE1-F1 score. To ensure the accuracy of the pairing, we reviewed 100 random pairings and found that this approach produced correct pairings in all cases.

3.1.4 English to Icelandic Translation

In order to augment our summarization data, we translated the XSum and CNN/DailyMail datasets from English to Icelandic using a machine translation model. Specifically, we utilized Facebook's multilingual model, which was a winning submission to the 2021 Conference on Machine Translation (WMT, Tran et al. (2021)). This model is fine-tuned for news domain data and trained using data from eight different languages, achieving state-of-the-art performance in machine translation. We used the pre-trained version of the model, which is available in HuggingFace's Transformers library, and loaded the weights from the `wmt21-dense-24-wide-en-x` repository. To improve the quality of translations, we split the text into sentences and translated them separately. This approach was found to improve translation quality during a manual inspection, although no quantitative evaluation was performed to confirm it.

3.2 Models

In this study, we introduce the Gullfaxi model, which is based on the PEGASUS architecture (Zhang et al., 2020) but trained on Icelandic text. We call the model $\text{Gullfaxi}_{\text{BASE}}$ and it corresponds to the BASE architecture presented in the PEGASUS study. $\text{Gullfaxi}_{\text{BASE}}$ has 223M trainable parameters. Additionally, we also fine-tune a pre-trained mT5 model (Xue et al., 2021) for performance comparison. We use mT5_{BASE}, which has 580M trainable parameters. The increase in parameter count compared to $\text{Gullfaxi}_{\text{BASE}}$ is primarily due to the larger vocabulary employed in mT5. Details on training and hyperparameters can be found in Appendix A

3.3 Downstream Tasks

We evaluate the performance of our models on a set of downstream summarization tasks using the RÚV Radio News (RRN) dataset. The RRN dataset is split into train, validation, and test sets with a 60%, 20%, 20% ratio respectively. We created three fine-tuning tasks to test different abilities for abstractive summarization:

Task 1: Intro + Main → Summary The task involves producing a summary from the introduction and main part of the story. As the introduction and summary are often similar and in some cases identical, this task is somewhat related to extractive summarization.

Task 2: Main → Intro The task involves generating the introduction from the main part of the story. The introduction and main text rarely share the same sentences, thus we expect the model to generate more abstractive summaries.

Task 3: Intro → Title The task involves producing the title of the story from the introduction. The title is much shorter compared to the output in the previous tasks, and we expect the model to generate more abstractive summaries.

To further understand the performance of Gullfaxi on larger corpora, we fine-tune it on the Icelandic translations of the XSum and CNN/DailyMail datasets and compare the results to the English PEGASUS model.

We also explored a mixed fine-tuning approach where the models were first fine-tuned on translated data and then on Icelandic data. For each fine-tuning phase, the model was fine-tuned until the validation loss stopped decreasing.

3.4 Performance Measures

In this study, we use the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scoring algorithm to evaluate the performance of our models (Lin, 2004). ROUGE is a widely used and accepted standard for evaluating automatic summarization tasks. We use ROUGE-1, ROUGE-2, and ROUGE-L to calculate the similarity between the model's summary and a reference summary.

We define $\text{count}_{\text{match}}(\text{gram}_n)$ as the number of matching n-grams, and similarly, $\text{count}_{\text{ref}}(\text{gram}_n)$ and $\text{count}_{\text{model}}(\text{gram}_n)$ refer to the number of n-grams in the reference and the model output, respectively. The ROUGE-N precision, recall, and F1-score are calculated as follows:

$$\text{ROUGE-N precision} = \frac{\text{count}_{\text{match}}(\text{gram}_n)}{\text{count}_{\text{model}}(\text{gram}_n)},$$

$$\text{ROUGE-N recall} = \frac{\text{count}_{\text{match}}(\text{gram}_n)}{\text{count}_{\text{ref}}(\text{gram}_n)}.$$

Similarly, we define ROUGE-L precision and recall using the longest common subsequence (LCS) between the reference summary and the model's output in the numerator. The LCS represents the longest sequence of words shared between the two texts, regardless of whether the words appear consecutively. Finally, we compute the F1-score for each of ROUGE-1, ROUGE-2, and ROUGE-L as the harmonic mean of their precision and recall.

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$

3.5 Human Evaluation

To further assess the quality of the generated summaries, we conduct human evaluations on a subset of the generated summaries. The samples are rated by a single annotator on three binary criteria: relevance, correctness, and language. Relevance is based on whether the summary is relevant to the reference text and pertains to the subject matter of the story. Correctness is based on whether the summary is factually accurate and consistent with the reference text, and does not include any unrelated information. Lastly, language is based on whether the summary is grammatically correct and natural, without any repetitions or use of non-Icelandic words.

4 Results

4.1 Summarization Performance

In this section, we present the results of our evaluation of the Gullfaxi model and the mT5 model on the RRN dataset. Table 2 shows the ROUGE F1-scores (R1/R2/RL) of the fine-tuned models for each task. The results show that mT5_{BASE} outperforms the Gullfaxi model on all tasks. The difference between the models is particularly notable in the first task (Intro + Main → Summary). Opting for an extractive approach in this task provides leverage in achieving high ROUGE scores as the intro and the summary tend to be similar. For comparison, a basic strategy of copying the intro yields a score of $61.8/46.2/58.9$. Examples of the model outputs and their scores can be found in Appendix B.

We also found that mT5 almost exclusively relied on an extractive approach in the first task, simply copying the intro, which resulted in a much higher score compared to Gullfaxi. In the other tasks, we observed more abstractive output from all models. Factors that contributed to lower ROUGE scores include repetition, grammatical errors, and different lengths of the output.

As a reference, we also fine-tuned a randomly initialized model, referred to as Transformer$_{\text{BASE}}$, with the same architecture as Gullfaxi$_{\text{BASE}}$ on the full RRN training set without any pre-training.

4.2 Low-resource Fine-tuning

In this section, we examine the performance of Gullfaxi and mT5 in a low-resource fine-tuning setting. We fine-tuned both models using varying amounts of data from the RRN dataset, specifically using the first 10^k $(k = 1, 2, 3)$ examples from the training set. Figure 1 show the results of the low-resource fine-tuning of Gullfaxi$_{\text{BASE}}$ and mT5$_{\text{BASE}}$.

Our findings indicate that even without fine-tuning, Gullfaxi$_{\text{BASE}}$ performed better than Transformer$_{\text{BASE}}$ on some tasks. mT5$_{\text{BASE}}$ also showed a gradual improvement in performance as the number of training examples increased. Both models achieved significantly higher scores than Transformer$_{\text{BASE}}$ when fine-tuned on the full RRN training set.

4.3 Fine-tuning Data Augmentation

In this section, we investigate the impact of data augmentation on fine-tuning Gullfaxi and mT5 for summarization tasks. Specifically, we fine-tune the models on the Icelandic translations of XSum and CNN/DailyMail datasets and evaluate their performance on the RRN dataset. We also explore an approach where the model is fine-tuned in two phases, first on augmented data and then on RRN data. Results are presented in Table 2. We observe that when the translations are combined with RRN, the scores are higher. Furthermore, by manually reviewing the output of the models, we notice an increase in grammatical errors when using the translations for fine-tuning for Gullfaxi but not for the mT5 model. To further demonstrate the difference in performance between Icelandic and English models trained in a similar manner, we evaluate the performance of the Gullfaxi model fine-tuned on XSum and CNN/DailyMail on their respective test sets. Table 3 shows the results with and without fine-tuning, as well as a comparison to the English data scores of the PEGASUS models obtained from the original study. It is apparent that fine-tuning leads to a notable improvement in performance on both datasets. However, when comparing Gullfaxi to PEGASUS, it is evident that the PEGASUS model's scores for English are much higher.

4.4 Human Evaluation

In order to further evaluate the performance of our models, we conducted a human evaluation of a subset of the summary outputs. We randomly sampled 50 examples from the Main $\rightarrow$ Intro task, which tests the model's ability to generate an abstractive summary in a few sentences. The results of the human evaluation are presented in Table 4, which compares the scores for Gullfaxi and mT5 for different fine-tuning approaches.

In general, we observed that the output intros produced by all models were of lower quality than those written by humans. The outputs were often relevant to the reference text but not effectively summarizing it. Fine-tuning Gullfaxi on the Icelandic translations of CNN/DailyMail resulted in the worst performance, particularly regarding grammar, often using the wrong inflections of words, as seen in Table 5.

We further observed that the mT5 model improved with augmented translation data, whereas Gullfaxi performed worse with the augmented data, particularly in grammar and word inflection. Overall, the mT5 model showed superior performance, producing the best summaries when fine-tuned on the augmentations followed by fine-tuning on the RRN dataset, demonstrating generalization to the summarization task. However, it sometimes extracted information from the reference text instead of generating new phrases, which may explain its higher scores for relevance and correctness compared to Gullfaxi.

5 Discussion

In this study, we investigated techniques for addressing the challenging task of low-resource abstractive summarization for Icelandic. We evaluated several well-known approaches and uncovered limitations as well as potential avenues for future work.

The main challenge in our study was the lack of sufficient data for training abstractive summarization models for Icelandic. We collected a news-domain abstractive summarization dataset, RÚV Radio News (RRN), but acknowledge that it is relatively small and may not generalize well to other domains or summarization settings. The collection and processing of RRN were time-consuming due to the inconsistency in the format of the radio transcripts. To aid in future language resource development, publicly funded organizations, such as

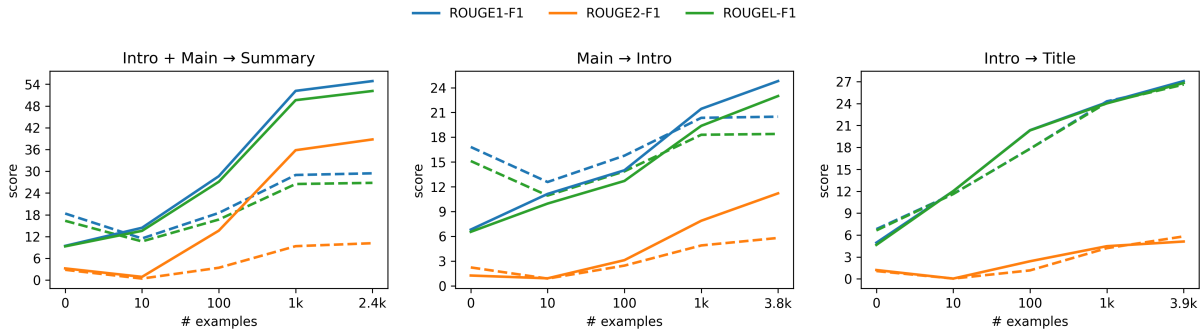


Figure 1: Model performance on RRN with a limited number of fine-tuning examples. The dashed lines are the performance of the Gullfaxi$_{\text{BASE}}$ model whereas the solid lines represent the mT5$_{\text{BASE}}$ model.

Model	**Intro + Main → Summary**	**Main → Intro**	**Intro → Title**
Transformer$_{\text{BASE}}$ (only fine-tuning)	13.1/1.3/12.0	10.8/0.7/9.7	14.2/0.6/14.1
Gullfaxi$_{\text{BASE}}$ (no fine-tuning)	18.4/2.9/16.4	16.8/2.2/15.1	6.8/1.1/6.6
Gullfaxi$_{\text{BASE}}$ (RRN)	29.4/10.2/26.8	20.5/5.7/18.5	26.6/5.8/26.6
Gullfaxi$_{\text{BASE}}$ (CNN/DailyMail)	26.5/9.0/24.2	17.8/3.5/16.0	-
Gullfaxi$_{\text{BASE}}$ (CNN/DailyMail + RRN)	42.5/21.3/39.6	22.2/6.2/19.7	-
mT5$_{\text{BASE}}$ (RRN)	54.9/38.8/52.1	24.8/11.2/23.0	27.1/5.1/26.8
mT5$_{\text{BASE}}$ (CNN/DailyMail)	36.2/19.3/33.9	21.4/5.9/19.4	-
mT5$_{\text{BASE}}$ (CNN/DailyMail + RRN)	**58.9/42.8/56.1**	**33.0/17.0/30.6**	-

Table 2: A comparison of Gullfaxi$_{\text{BASE}}$ and mT5$_{\text{BASE}}$ on the RRN dataset using different training sets. Transformer$_{\text{BASE}}$ has the same model architecture as Gullfaxi$_{\text{BASE}}$ but is not pre-trained, only randomly initialized. The scores listed are the ROUGE F1-scores (R1/R2/RL). The information in brackets denotes what data the model was fine-tuned on, when fine-tuned on more than a single dataset, the training is performed in two phases. Highest scores in the first two columns are shown in bold.

Model	**XSum$_{\text{is}}$**	**CNN/DailyMail$_{\text{is}}$**
Gullfaxi$_{\text{BASE}}$ (no fine-tuning)	13.3/1.1/11.4	13.1/1.3/12.1
Gullfaxi$_{\text{BASE}}$ (XSum)	23.5/7.3/19.9	-
Gullfaxi$_{\text{BASE}}$ (CNN/DailyMail)	-	24.6/7.7/23.1
	XSum$_{\text{en}}$	**CNN/DailyMail$_{\text{en}}$**
PEGASUS$_{\text{BASE}}$	39.8/16.6/31.7	41.8/18.8/38.9

Table 3: Gullfaxi$_{\text{BASE}}$'s ROUGE F1-scores (R1/R2/RL) with and without fine-tuning on the Icelandic translations of XSum and CNN/DailyMail. The scores listed for PEGASUS$_{\text{BASE}}$ are the highest English language fine-tuning scores obtained from the PEGASUS paper.

RÚV, should be encouraged to be more mindful of their data processing. RRN provides a starting point for further research in this field, but broader coverage and diversity are necessary to create practical summarization solutions for Icelandic.

We evaluated the performance of two Gullfaxi models and mT5 on the RRN dataset for abstractive summarization in a low-resource setting, specifically for the Icelandic language. mT5 consistently outperformed the Gullfaxi models. However, we also observed that pre-training Gullfaxi led to better summarization performance when compared to no pre-training.

The performance of the multilingual mT5 model can be attributed to the large corpus of multilingual data, including 2.1 million Icelandic pages, used for pre-training. It should further

Model	Relevance	Correctness	Language
Gullfaxi$_{\text{BASE}}$ (RRN)	74%	8%	42%
Gullfaxi$_{\text{BASE}}$ (CNN/DailyMail)	46%	8%	6%
Gullfaxi$_{\text{BASE}}$ (CNN/DailyMail + RRN)	64%	4%	10%
mT5$_{\text{BASE}}$ (RRN)	84%	46%	54%
mT5$_{\text{BASE}}$ (CNN/DailyMail)	80%	42%	44%
mT5$_{\text{BASE}}$ (CNN/DailyMail + RRN)	**96**%	**54**%	**56**%

Table 4: The human evaluation scores for the Main → Intro task. The scores listed are the fraction of summary results that fulfilled the criteria of each category in the 50 annotations evaluated. Highest scores in each column are shown in bold.

Reference summary	Framkvæmdastjóri Vistorku á Akureyri segir raunsæjan kost að Ísland geti orðið algjörlega óháð olíu á næstu árum og þar með sjálfbært um alla orkuframleiðslu. Heildræna stefnu vanti þó í málaflokknum.
Model output	framkvæmdastjóri segir að það **er engin** (séu engar) hindranir **til** (fyrir því) að flýta banni við innflutningi á olíu
Reference summary	Í fyrsta sinn í 15 ár er stefnt að því að byggja fjölda íbúðarhúsa norður af Akureyri. Mikill áhugi er á lóðunum og færri fengu úthlutun en vildu.
Model output	á síðustu 19 árum **hafa lóðir** (hefur lóðum) verið úthlutað til eldri borgara. áhugi á **lóðir** (lóðum) hefur aukist á undanförnum árum.

Table 5: Examples of ungrammatical output text of Gullfaxi$_{\text{BASE}}$ fine-tuned exclusively on the Icelandic translations of CNN/DailyMail. Corrections are in parentheses. The inflections of the words in red are incorrect.

benefit from the translation task, which is one of the tasks it is trained on in the pre-training phase. Our results suggest that low-resource languages may benefit from the general knowledge acquired through multilingual pre-training when fine-tuned for specific tasks, aligning with previous work (Snæbjarnarson et al., 2022; Snæbjarnarson and Einarsson, 2022a) where multilingual models for Icelandic were studied. For Gullfaxi, we used the same hyperparameters as in the Pegasus paper, but we still cannot conclude that Gullfaxi cannot be made better since we did not perform extensive hyperparameter tuning of the model due to time and cost.

We explored using machine translations to augment data for low-resource NLP tasks, specifically abstractive summarization in Icelandic. We fine-tuned models on Icelandic translations of two large English summarization datasets, CNN/DailyMail and XSum, but found the fine-tuned model did not perform well on the Icelandic summarization task, RÚV Radio News (RRN) and had more grammar mistakes compared to other models. When reviewing the Icelandic translations used for data augmentation there are a few things to note. Although most of them are easily comprehensible for a native speaker, they tend to be unnatural, use unusual wording, and have the wrong inflection of words. For that reason, we think that exclusively using translated examples for fine-tuning can sometimes lead to worse output texts.

We also explored a two-phase fine-tuning approach where we first fine-tuned on translated data and then on RRN. We observed improvements in ROUGE metrics but a manual inspection revealed better summaries for the mT5 model but worse summaries for the Gullfaxi model when compared to using no augmentation. This difference highlights the limitation of using ROUGE scores as a metric to measure summarization performance. It further highlights the importance of the quality of training data in low-resource settings, as well as the importance of considering the naturalness and grammatical accuracy of machine translations when using them for data augmentation.

We investigated the impact of the number of fine-tuning examples on the performance of a low-resource abstractive summarization task by fine-tuning $\text{Gullfaxi}_{\text{BASE}}$ with 0, 10, 100, and 1k examples from the RRN dataset. Our results showed that even without fine-tuning, $\text{Gullfaxi}_{\text{BASE}}$ performed better on some tasks than a randomly initialized model fine-tuned on all the RRN training set examples. As we increased the number of fine-tuning examples, $\text{Gullfaxi}_{\text{BASE}}$ continued to improve, achieving significantly higher scores than the baseline when using the full RRN dataset. This demonstrates the effectiveness of pre-training in a low-resource setting and highlights the potential value of creating small, domain-specific summarization datasets. However, we also observed that the mT5 model was better able to make use of more fine-tuning examples when exceeding a thousand examples.

The study has several limitations, including that all models tend to generate summaries that are inconsistent with the source text, which is a common issue for abstractive summarization models, and limits their practical use (Cao et al., 2018; Bender et al., 2021). Another limitation is that the pre-training objective may encourage the generation of incorrect statements. To address this, the use of reinforcement learning with human feedback, as demonstrated by the Instruct GPT model (Ouyang et al., 2022) can be used. Additionally, it is worth noting that most state-of-the-art models and breakthrough studies in NLP are primarily focused on English-language solutions, and it is unclear to what extent the choice of language impacts the performance of these models when the training budget and amount of training data are fixed. Further research comparing the performance of state-of-the-art models across different languages would be necessary to better understand this issue. Lastly, we would like to highlight the potential of including the Main $\rightarrow$ Summary task in future research, which was deemed out of scope for this work.

Our evaluation approach may be perceived as a limitation due to its binary nature. However, we intentionally designed it this way to prioritize objectivity, by being stringent about any errors in the model-generated summaries. That said, there are alternative evaluation approaches that could be explored in future research, such as employing continuous rating scales or more nuanced assessment criteria to better capture the intricacies of summary quality. By investigating these alternatives, we can potentially gain a deeper understanding of the strengths and weaknesses of summarization models.

Our results on data augmentation show that evaluating abstractive summaries is challenging. In this study, we used ROUGE scores and human evaluation, but ROUGE has been known to favor lexical similarity, which may not be suitable for abstractive summaries (Ng and Abrecht, 2015), particularly in morphologically rich languages like Icelandic. The lower ROUGE scores of Icelandic summaries compared to English language studies may be due to the differences in grammar between the two languages. The human evaluation revealed that the summaries tended to be factually inaccurate and had varying levels of grammatical quality. For future evaluations, it could be beneficial to include human-written summaries for comparison.

6 Conclusion

In this work, we explored methods for automatic abstractive summarization in a low-resource setting, specifically in the Icelandic language. We collected and published the first publicly available abstractive summarization dataset for Icelandic, and used it to train and evaluate state-of-the-art models. Our findings indicate that multilingual pre-training provides significant benefits for this task, as the multilingual mT5 model consistently outperformed a similar capacity PEGASUS model pre-trained from scratch on Icelandic text only. Additionally, we found that using machine translations for data augmentation led to higher ROUGE scores. However, when evaluated manually, the benefits of data augmentation were not consistently observed across models compared to a scenario where models were solely fine-tuned on the RRN dataset. Specifically, data augmentation enhanced the quality of the summaries generated by the mT5 model compared to those produced with RRN fine-tuning alone. In contrast, the Gullfaxi model's summaries experienced a decrease in quality due to data augmentation, displaying weaker grammar and a higher level of inconsistency compared to the reference text.

For future work, we suggest a further collection of abstractive summarization data for Icelandic, as well as studying metrics that may be better suited for this language. We also emphasize the benefits

of using pre-trained multilingual models, which we expect to apply to other generative tasks and languages. Overall, our study highlights the importance of pre-training and the challenges of evaluating abstractive summarization in low-resource settings.

Acknowledgments

We thank Prof. Dr.-Ing. Morris Riedel and his team for providing access to the DEEP supercomputer at Forschungszentrum Jülich.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 610–623, New York, NY, USA. Association for Computing Machinery. 8

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32. 8

Karin Christiansen. 2014. *Summarization of Icelandic Texts*. Ph.D. thesis, Master's thesis, Reykjavik University, Reykjavik, Iceland. 2

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics. 2

Jón Daðason, Hrafn Loftsson, Salome Sigurðardóttir, and Þorsteinn Björnsson. 2021. IceSum: An Icelandic Text Summarization Corpus. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 9–14, Online. Association for Computational Linguistics. 2

Jón Friðrik Daðason and Hrafn Loftsson. 2022. Pretraining and Evaluating Transformer-based Language Models for Icelandic. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7386–7391, Marseille, France. European Language Resources Association. 2

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics. 2

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics. 2

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc. 2

Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. 2022. Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*. 1

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics. 11

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics. 2

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics. 4

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics. 2

Jun-Ping Ng and Viktoria Abrecht. 2015. Better summarization evaluation with word embeddings for ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1925–1930, Lisbon, Portugal. Association for Computational Linguistics. 8

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*. 8

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551. 2

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics. 2

Haukur Barri Símonarson, Vésteinn Snæbjarnarson, Pétur Orri Ragnarson, Haukur Jónsson, and Vilhjalmur Thorsteinsson. 2021. Miðeind's WMT 2021 submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 136–139, Online. Association for Computational Linguistics. 2

Vésteinn Snæbjarnarson, Haukur Barri Símonarson, Pétur Orri Ragnarsson, Svanhvít Lilja Ingólfsdóttir, Haukur Jónsson, Vilhjalmur Thorsteinsson, and Hafsteinn Einarsson. 2022. A warm start and a clean crawled corpus - a recipe for good language models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4356–4366, Marseille, France. European Language Resources Association. 2, 7

Vésteinn Snæbjarnarson and Hafsteinn Einarsson. 2022a. Cross-Lingual QA as a Stepping Stone for Monolingual Open QA in Icelandic. In *Proceedings of the Workshop on Multilingual Information Access (MIA)*, pages 29–36, Seattle, USA. Association for Computational Linguistics. 2, 7

Vésteinn Snæbjarnarson and Hafsteinn Einarsson. 2022b. Natural Questions in Icelandic. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4488–4496, Marseille, France. European Language Resources Association. 2

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936. PMLR. 2

Steinþór Steingrímsson, Sigrún Helgadóttir, Eiríkur Rögnvaldsson, Starkaður Barkarson, and Jón Guðnason. 2018. Risamálheild: A Very Large Icelandic Text Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA). 2

Martin A. Tanner and Wing Hung Wong. 1987. The Calculation of Posterior Distributions by Data Augmentation. *Journal of the American Statistical Association*, 82(398):528–540. Publisher: Taylor & Francis. 1

Oguzhan Tas and Farzad Kiyani. 2017. A survey automatic text summarization. *PressAcademia Procedia*, 5(1):205–213. 1

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook AI's WMT21 news translation task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 205–215, Online. Association for Computational Linguistics. 3

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. 1, 2

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics. 2, 3

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc. 2

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR. ISSN: 2640-3498. 1, 2, 3

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2021. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1):43–76. Conference Name: Proceedings of the IEEE. 1

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics. 1

A Training details

A.1 Pre-training Objective

Gullfaxi is pre-trained using a self-supervised pre-training objective called gap sentence generation

(GSG). This method, originally proposed for the PEGASUS model, involves masking whole sentences from the input document and concatenating them in their original order to form an abstractive summary-like output text. The goal is that a pre-training objective that closely resembles the summarization task will lead to a better starting point for fine-tuning.

The gap sentences are selected using a heuristic approach based on importance criteria. The ideal outcome is that the sentences containing the key information of the text are chosen from the document, but this is not guaranteed by the method. The importance of a sentence within a document is estimated by calculating the ROUGE1-F1 score between the gap sentence and the rest of the document. In this study, we calculate this score based on the lemmatized sentences as they are given in the IGC, due to the inflected nature of the Icelandic language.

The highest-performing models from the PEGASUS study were obtained by choosing a gap sentence ratio between 15%-45%, varying by task and model. For this study, we mask 20% of the total number of sentences in the original text document.

A.2 Vocabulary

The vocabulary of a language model is the set of unique subword units, referred to as tokens, that the model is able to recognize. Methods such as PEGASUS and mT5 construct the vocabulary by training a subword tokenizer, which aims to identify an appropriate separation of input text. In this study, we use the SentencePiece unigram tokenizer (Kudo and Richardson, 2018) to construct a vocabulary for Gullfaxi. This configuration is similar to that used in PEGASUS, with a vocabulary size of 96k and no differentiation between lowercase and uppercase letters. The Gullfaxi tokenizer is trained on documents from the Icelandic Gigaword Corpus (IGC). On the other hand, the mT5 model comes with a pre-trained multilingual vocabulary of size 250k, obtained from training a SentencePiece tokenizer on the mC4 dataset.

A.3 Hyperparameter configuration

In this study, we use the same hyperparameter configuration as the PEGASUS model, as it is computationally expensive to train and conduct a search for optimal hyperparameters. Details of the experiments' hyperparameters and training configuration can be found in the appendix. The Gullfaxi model is trained from scratch and implemented using the HuggingFace Transformers library, while the mT5 model uses pre-trained weights from the `google/mt5-base` repository and the corresponding tokenizer. The training was conducted on a high-performance computing cluster using multiple GPUs and distributed configuration with the HuggingFace Accelerate library. During fine-tuning, we use a label-smoothed regularization with a value of 0.1, and at test time, we use a beam size of 8 with a length penalty of 0.8 for all tasks.

B Example model outputs

Examples of model output can be seen in Tables 7, 8, and 9.

Pre-training of Gullfaxi				
Model	**# Steps**	**Batch size**	**Max input tokens**	**Max target tokens**
$\text{Gullfaxi}_{\text{BASE}}$	200k	256	512	256
Fine-tuning of Gullfaxi models in Table 2, 3 and 4				
Task	**# Steps**	**Batch size**	**Max input tokens**	**Max target tokens**
Intro + Main $\rightarrow$ Summary	4k	256	512	128
Main $\rightarrow$ Intro	4k	256	512	128
Intro $\rightarrow$ Title	4k	256	128	32
XSum	50k	256	512	64
CNN/DailyMail	50k	256	512	128
Fine-tuning of mT5_{BASE} in Table 2 and 4				
Task	**# Steps**	**Batch size**	**Max input tokens**	**Max target tokens**
RRN	8k	256	Same as Gullfaxi	
Low-resource fine-tuning of $\text{Gullfaxi}_{\text{BASE}}$ and mT5_{BASE} in Figure 1				
Task	**# Steps**	**Batch size**	**Max input tokens**	**Max target tokens**
RRN	3k	256	Same as Gullfaxi	

Table 6: Hyperparameter setup for pre-training and fine-tuning.

	RRN Document	
Title	Neyðarástand vegna flóða í Kína (e. Emergency due to floods in China)	
Intro	Hæsta viðbúnaðarstigi hefur verið lýst yfir í Henan-héraði í Kína vegna flóða. Þau hafa orðið að minnsta kosti tólf manns að bana. (e. The highest level of preparedness has been declared in Henan province in China due to floods. At least twelve people have been killed.	
Main	Hátt í tvö hundruð þúsund íbúar borgarinnar Sheng-sjá Zhengzhou í Henan-héraði í Kína hafa verið fluttir að heiman vegna flóða. Þau hafa orðið að minnsta kosti tólf manns að bana. Hæsta viðbúnaðarstigi hefur verið lýst yfir í héraðinu. Ríkisfjölmiðlar í Kína hafa eftir Xi Jinping að ástandið í Henan sé afar alvarlegt. Stíflur hafi brostið og valdið manntjóni og eignatapi. Allir verði að leggjast á árarnar til að koma í veg fyrir að það verði enn meira. Á annan tug borga og bæja eru umflotin vatni. Á götum hafa myndast straumharðar ár sem bera með sér bíla og alls kyns brak. Ef marka má fréttir af svæðinu er ástandið verst í héraðshöfuðborginni Zhengzhou. Þar fræddi vatn inn í jarðlestagöng með þeim afleiðingum að tólf drukknuðu. Um fimm hundruð var bjargað úr göngunum. Hátt í tvö hundruð þúsund íbúum borgarinnar hefur verið forðað að heiman vegna flóða. Síðustu þrjá sólarhringa hefur fallið álíka mikið regn og á einu ári. Þá greindi kínverski herinn frá því í gær að tuttugu metra sprunga væri komin í Yihetan stífluna í Luoyang þar sem um það bil sjö milljónir búa. Hermenn hafa verið sendir á vettvang til að stýra rennsli í ám og hlaða upp sandpokum til að styrkja bakkana. Vegna veðursins í miðhluta Kína hefur fjölda flugferða verið aflýst og áætlunarferðir járnbrautarlesta eru úr skorðum. (e. Around two hundred thousand residents of the city of Zhengzhou in Henan province in China have been evacuated from their homes due to floods. The floods have killed at least twelve people. The highest preparedness level has been declared in the province. State media in China have quoted Xi Jinping as saying that the situation in Henan is very serious. Dams have burst, causing casualties and property damage. Everyone must do their part to prevent further damage. Between ten and twenty cities and towns are flooded with water. On the streets, strong currents have formed, carrying cars and various debris with them. Judging by news from the area, the situation is worst in the provincial capital of Zhengzhou. There, water flooded into subway tunnels, resulting in twelve people drowning. About five hundred people were rescued from the tunnels. Almost two hundred thousand inhabitants of the city have been forced from their homes due to the floods. In the past three days, as much rain has fallen as in an entire year. The Chinese army reported yesterday that a twenty-meter crack has appeared in the Yihetan dam in Luoyang, where around seven million people live. Soldiers have been sent to the site to control the flow of rivers and build sandbags to strengthen the banks. Due to the weather in central China, numerous flights have been canceled and scheduled train trips are disrupted.)	
Summary	Hæsta viðbúnaðarstigi hefur verið lýst yfir í Henan héraði í Kína vegna flóða. Að minnsta kosti tólf manns hafa látist (e. The highest level of preparedness has been declared in Henan province in China due to floods. At least twelve people have died.).	
	Gullfaxi$_{\text{BASE}}$ (RRN)	**ROUGE F1**
Title	skjálfti í henan-héraði	25.00/0.00/25.00
Intro	að minnsta kosti fimm eru látnir eftir að öflugur jarðskjálfti reið yfir kína í gær. R / C / L : $1/0/1$ *The text is relevant to the topic of a natural disaster in China but incorrectly refers to the cause of the disaster as an earthquake. The language used is natural and grammatically correct.*	35.29/11.43/29.41
Summary	minnst níu hafa látist í miklum flóðum í kína frá því í gær. tugir þúsunda hafa orðið að yfirgefa heimili sín vegna flóða.	35.00/14.29/30.00
	Gullfaxi$_{\text{BASE}}$ (CNN/DailyMail)	**ROUGE F1**
Intro	að minnsta kosti 12 manns hafa verið drepnir í henan héraði í henan-héraði í kína R / C / L : $0/0/0$ *The text is not relevant to the topic, incorrectly capturing the subject of the story, asserting that people were murdered in China. The language used includes unnatural repetitions.*	54.55/28.57/48.48
Summary	að minnsta kosti 12 manns hafa látist í henan héraði í henan-héraði í henan-héraði í kína	64.52/50.00/64.52
	Gullfaxi$_{\text{BASE}}$ (CNN/DailyMail + RRN)	**ROUGE F1**
Intro	hátt í tvö þúsund íbúar í kína hafa orðið að yfirgefa heimili sín vegna flóða í henan-héraði í kína. R / C / L : $1/0/1$ *The text is relevant but contains a factual error regarding the number of people who left their homes due to the flood. The language used is natural and grammatically correct.*	45.71/31.58/40.00
Summary	hæsta viðbúnaðarstigi hefur verið lýst yfir í kína vegna flóða í henan-héraði í kína vegna flóða. þau hafa orðið að minnsta kosti tólf að bana í henan-héraði í kína.	78.95/58.54/73.68
	mT5$_{\text{BASE}}$ (RRN)	**ROUGE F1**
Title	Hæsta viðbúnaðarstig lýst yfir í Henan	18.18/0.00/18.18
Intro	Síðustu þrjá sólarhringa hefur fallið ástandið verst í miðhluta Kína. Hátt í tvö hundruð þúsund íbúar borgarinnar hafa verið fluttir að heiman vegna flóða. R / C / L : $1/0/0$ *The text is relevant but the first sentence is unclear and unnatural.*	37.21/4.55/32.56
Summary	Hæsta viðbúnaðarstigi hefur verið lýst yfir í Henan-héraði í Kína vegna flóða. Þau hafa orðið að minnsta kosti tólf manns að bana.	82.05/65.00/76.92
	mT5$_{\text{BASE}}$ (CCN/DailyMail)	**ROUGE F1**
Intro	Hátt í tvö hundruð þúsund íbúar borgarinnar Sheng-sjá Zhengzhou í Henan-héraði í Kína hafa verið fluttir að heiman vegna flóða R / C / L : $1/1/1$ *The text meets all criteria but note that it copies the first sentence of the input text.*	42.11/20.00/36.84
Summary	Hæsta viðbúnaðarstig hefur verið lýst yfir í Henan-héraði í Kína vegna flóða	60.00/46.67/60.00
	mT5$_{\text{BASE}}$ (CCN/DailyMail + RRN)	**ROUGE F1**
Intro	Flóðin í Henan-héraði í Kína hafa orðið að minnsta kosti tólf að bana. Stíflur hafa brostið og valdið manntjóni og eignatapi. R / C / L : $1/1/1$ *The text meets all criteria. The first sentence effectively rephrases information from the input text while the second sentence is a direct copy.*	54.05/43.90/54.05
Summary	Hæsta viðbúnaðarstigi hefur verið lýst yfir í Henan-héraði í Kína vegna flóða. Þau hafa orðið að minnsta kosti tólf manns að bana.	82.05/65.00/76.92.

Table 7: Example of model output. English translations are shown for the source text and the human evaluation scores are shown for each translation along with a justification.

	RRN Document	
Title	Stríðið gæti varað árum saman	
Intro	Stríðið í Úkraínu gæti varað árum saman og Vesturlönd þurfa að vera undir það búin að veita Úkraínu stuðning áfram. Þetta segir Jens Stoltenberg framkvæmdastjóri Atlantshafsbandalagsins.	
Main	Stoltenberg lét þessi orð falla í viðtalið við þýska dagblaðið Bild. Hann segir að þrátt fyrir að stríðið sé kostnaðarsamt verði Vesturlönd að halda áfram að senda Úkraínumönnum hergögn og nauðsynjar því kostnaðurinn verði umtalsvert meiri ef Vladimír Pútín Rússlandsforseti nái sínum markmiðum. Við þurfum að búa okkur undar það að stríðið gæti varað árum saman, segir Stoltenberg. Það sama segir Boris Johnson, forsætisráðherra Bretlands, sem fór í sína aðra heimsókn til Kænugarðs á föstudag. "It would be a catastrophe if Putin won. It would be a catastrophe if he was able to secure the land bridge to the cities in the south that he has, to hold the Donbas. That's what he wants. Boris Johnson segir skelfilegt að hugsa til þess að Pútín vinni stríðið. Stuðningur við Úkraínu sé nauðsynlegur til að koma í veg fyrir að Rússar nái Donbas. Breska varnarmálaráðuneytið birti í morgun yfirferð um átökin sem geisa hvað harðast við borgina Sjevjerodonetsk. Þar kemur fram að litlar breytingar hafi orðið síðasta sólarhringinn. Síðustu daga hafi verið nokkuð um liðhlaup úr úkraínskum hersveitum. Einnig segir breska varnarmálaráðuneytið að átök haldi áfram innan hersveita Rússa og dæmi séu um að hersveitir neiti að hlýða skipunum foringja sinna.	
Summary	Framkvæmdastjóri Atlandshafsbandalagsins segir að Vesturlönd þurfi að vera undir það búin að veita Úkraínu stuðning áfram, stríðið geti varað í mörg ár í viðbót.	
	Gullfaxi$_{\text{BASE}}$ (RRN)	**ROUGE F1**
Title	jens stoltenberg um úkraínu	0.00/0.00/0.00
Intro	vladimír pútín rússlandsforseti segir að stríðið í úkraínu sé í fullum gangi. pútín rússlandsforseti segir að rússar verði að sætta sig við stríðið í úkraínu.	25.00/9.09/25.00
Summary	framkvæmdastjóri atlantshafsbandalagsins segir að stríðið í úkraínu sé hafið. framkvæmdastjóri atlantshafsbandalagsins segir að stríðið í úkraínu sé hafið.	40.00/6.25/33.33
	Gullfaxi$_{\text{BASE}}$ (CNN/DailyMail)	**ROUGE F1**
Intro	forsætisráðherra bretlands segir að stríðið í úkraínu sé nauðsynlegurnauðsynlegur til að koma í veg fyrir að rússland nái árangri í úkraínu	25.00/9.09/25.00
Summary	jens stoltenberg, framkvæmdastjóri atlantshafsbandalagsins í úkraínu, segir að vladimírs pútín rússlandsforseti muni halda áfram að senda hergögn til kænugarðs. forsætisráðherra bretlands, donbas, segir að rússneskir hermenn hafi verið sendir til kænugarðs til að koma í veg fyrir átök í úkraínu.	19.23/3.33/15.38
	Gullfaxi$_{\text{BASE}}$ (CNN/DailyMail + RRN)	**ROUGE F1**
Intro	forsætisráðherra bretlands segir skelfilegt að stríðið í úkraínu sé að undirbúa sig fyrir það. hann segir að stríð í úkraínu sé á réttri leið.	28.57/8.70/28.57
Summary	jens stoltenberg knattspyrnustjóri atlantshafsbandalagsins segir að stríðið í úkraínu gæti komið í veg fyrir að rússar nái friði í úkraínu.	26.32/4.76/21.05
	mT5$_{\text{BASE}}$ (RRN)	**ROUGE F1**
Title	Stríðið í Úkraínu	25.00/0.00/25.00
Intro	Forsætisráðherra Bretlands segir skelfilegt að hugsa til þess að Vladimír Pútín Rússlandsforseti nái sínum markmiðum.	10.53/0.00/10.53
Summary	Forsætisráðherra Bretlands segir að stríðið í Úkraínu gæti varað árum saman og Vesturlönd þurfa að vera undir það búin að veita Úkraínu stuðning áfram. Þetta segir framkvæmdastjóri Atlantshafsbandalagsins.	62.22/36.00/44.44
	mT5$_{\text{BASE}}$ (CCN/DailyMail)	**ROUGE F1**
Intro	Stuðningur við Úkraínu er nauðsynlegur til að koma í veg fyrir að Rússar nái Donbas segir Boris Johnson forsætisráðherra Bretlands	18.60/0.00/13.95
Summary	Stuðningur við Úkraínu er nauðsynlegur til að koma í veg fyrir að Rússar nái Donbas segir Jens Stoltenberg framkvæmdastjóri Atlantshafsbandalagsins	25.00/0.00/10.00
	mT5$_{\text{BASE}}$ (CCN/DailyMail + RRN)	**ROUGE F1**
Intro	Forsætisráðherra Bretlands segir skelfilegt að hugsa til þess að Pútín Rússlandsforseti vinni stríðið. Stuðningur við Úkraínu sé nauðsynlegur til að koma í veg fyrir að Rússar nái Donbas.	20.83/0.00/12.50
Summary	Stríðið í Úkraínu gæti varað árum saman og Vesturlönd þurfa að vera undir það búin að veita Úkraínu stuðning áfram, segir framkvæmdastjóri Atlantshafsbandalagsins.	69.77/39.13/46.51.

Table 8: Example of model output.

	RRN Document	
Title	Breytingar á leigumarkaði	
Intro	Ungt fólk hefur hrakist af leigumarkaði í covid-faraldrinum og hefur í vaxandi mæli þurft að flytja aftur heim í foreldrahús. Vísbendingar eru um að dregið hafi úr framboði á leiguhúsnæði á síðustu mánuðum.	
Main	Þetta kemur fram í könnun á vegum hagdeildar Húsnæðis- og mannvirkjastofnunar um stöðuna á leigumarkaði. Almennt hefur leigjendum fækkað á síðustu tveimur árum. Margir hafa nýtt sér hagstæð lánakjör til að kaupa húsnæði en aðrir hafa hrakist af markaðinum. Karlotta Halldórsdóttir hagfræðingur hjá stofnuninni segir að ungt fólk hafi orðið hvað verst úti. Svo myndi ég segja líka að áhrif af covid faraldrinum þau eru að koma einna verst niður á búsetu hjá ungu fólki. Yngsta aldurshópnum 18 til 24 ára. Þar erum við að sjá stórlækkun hjá þeim á leigumarkaði og þeir eru í auknum mæli að flytja aftur í foreldrahús Við mundum geta giskað á að það væri minni atvinna fyrir þau. Kannski mikið til fólk sem er í hlutastörfum, veitingageiranum, ferðaþjónustutengt og atvinna þar hafi minnkað og þau séu með minna ráðstöfunarfé og hafi þá ekki lengur efni á að vera á leigumarkaði Þá eru einnig vísbendingar um að dregið hafi úr framboði á leiguhúsnæði á undanförnum mánuðum. Í upphafi faraldursins jókst framboð verulega þegar Airbnb íbúðir fóru inn á hinn almenna leigumarkað. Við sáum einmitt í kjölfarið af covid og í fyrra þá vorum við að sjá framboð af leiguhúsnæði aukast töluvert. Fólk átti auðveldara með að finna sér húsnæði og svona aukið öryggi. En við erum að sjá í fyrsta skipti núna síðan 2015 að það sé hærra hlutfall sem finnst erfiðara að verða sér úti um húsnæði. Þannig að það eru vísbending um að það sé að draga úr þessu mikla framboði sem var Þetta hefur líka þær afleiðingar að leiguverð fer nú hækkandi. Við sáum það í fyrsta skipti í svolítinn tíma þá hækkaði vísitalan. alveg töluvert milli mánaða. þannig að það eru kannski svona einhver merki um að það sé aðeins farið að vera erfiðari leigumarkaðurinn hjá okkur	
Summary	Staða ungs fólks á húsleigumarkaði hefur versnað í Covid faraldrinum og margir hafa neyðst til að flytja aftur í foreldrahús. Þetta sýnir ný könnun.	
	Gullfaxi$_{\text{BASE}}$ (RRN)	**ROUGE F1**
Title	skortur á leiguhúsnæði	33.33/0.00/33.33
Intro	ungtungt fólk á aldrinum 18 til átján ára hefur flutt lögheimili sitt á almennum leigumarkaði. þetta kemur fram í nýrri könnun.	20.83/0.00/20.83
Summary	ungt fólk hefur dregið úr framboði á leigumarkaði frá því í fyrra. ungt fólk er líklegra til að flytjast búferlum frá heimilum sínum.	26.67/4.35/22.22
	Gullfaxi$_{\text{BASE}}$ (CNN/DailyMail)	**ROUGE F1**
Intro	ungt fólk á aldrinum 18 til 24 ára hefur fækkað á undanförnum tveimur árum. ungt fólk á aldrinum 18 til 24 ára eru verst úti á leigumarkaði.	26.67/3.92/22.22
Summary	ungt fólk hefur dregið úr framboði á íbúðarhúsnæði á undanförnum mánuðum. á undanförnum mánuðum hefur fjöldi leigjenda dregið úr framboði á leiguhúsnæði á undanförnum mánuðum.	10.81/0.00/10.81
	Gullfaxi$_{\text{BASE}}$ (CNN/DailyMail + RRN)	**ROUGE F1**
Intro	ungt fólk á aldrinum 18 til 24 ára er í auknum mæli á leigumarkaði í fyrsta sinn í 20 ár. þetta kemur fram í nýrri könnun sem gerð var á vegum húsnæðis- og mannvirkjastofnunar sem unnin var á dögunum.	23.73/5.80/20.34
Summary	ungt fólk hefur hrakist af leigumarkaði í vaxandi mæli á síðustu mánuðum. vísbendingar eru um að ungt fólk hafi dregið úr framboði covid-faraldrinum.	21.74/0.00/17.39
	mT5$_{\text{BASE}}$ (RRN)	**ROUGE F1**
Title	Leigumarkaður	0.00/0.00/0.00
Intro	Forstjóri Húsnæðis- og mannvirkjastofnunar segir að ungt fólk hafi orðið hvað verst úti um húsnæði. Þetta kemur fram í könnun á vegum hagdeildar Húsnæðis- og mannvirkjastofnunar um stöðuna á leigumarkaði.	33.96/3.39/18.87
Summary	Formaður Húsnæðis- og mannvirkjastofnunar segir að ungt fólk hefur hrakist af leigumarkaði í covid-faraldrinum og hefur í vaxandi mæli þurft að flytja aftur heim í foreldrahús. Vísbendingar eru um að dregið hafi úr framboði á leiguhúsnæði á síðustu mánuðum.	28.57/9.68/25.00
	mT5$_{\text{BASE}}$ (CCN/DailyMail)	**ROUGE F1**
Intro	Almennt hefur leigjendum fækkað á undanförnum tveimur árum. Margir hafa nýtt sér hagstæð lánakjör til að kaupa húsnæði. En aðrir hafa hrakist af markaðinum.	19.61/3.64/19.61
Summary	Almennt hefur leigjendum fækkað á síðustu tveimur árum. Margir hafa nýtt sér hagstæð lánakjör til að kaupa húsnæði. En aðrir hafa hrakist af markaðinum.	25.53/8.51/21.28
	mT5$_{\text{BASE}}$ (CCN/DailyMail + RRN)	**ROUGE F1**
Intro	Ungt fólk er í auknum mæli að flytja aftur í foreldrahús. Þetta segir hagfræðingur hjá Húsnæðis- og mannvirkjastofnun. Almennt hefur leigjendum fækkað á síðustu tveimur árum.	45.28/17.54/45.28
Summary	Ungt fólk hefur hrakist af leigumarkaði og hefur í vaxandi mæli þurft að flytja aftur heim í foreldrahús. Vísbendingar eru um að dregið hafi úr framboði á leiguhúsnæði á síðustu mánuðum.	34.62/10.91/30.77.

Table 9: Example of model output.

ASR Language Resources for Faroese

Carlos Mena
Language and Voice Lab,
Reykjavík University,
102 Reykjavík, Iceland
carlosm@ru.is

Annika Simonsen
University of Iceland,
107 Reykjavík, Iceland
annika@hi.is

Jón Guðnason
Language and Voice Lab,
Reykjavík University,
102 Reykjavík, Iceland
jg@ru.is

Abstract

The aim of this work is to present a set of novel language resources in Faroese suitable for the field of Automatic Speech Recognition including: an ASR corpus comprised of 109 hours of transcribed speech data, acoustic models in systems such as WAV2VEC2, NVIDIA-NeMo, Kaldi and PocketSphinx; a set of n-gram language models and a set of pronunciation dictionaries with two different variants of Faroese. We also show comparison results between the distinct acoustic models presented here. All the resources exposed in this document are publicly available under creative commons licences.

1 Introduction

As the digital world has become increasingly prominent and omnipresent in most human activities, the use of more and better language technologies has become a pressing need. For this reason, more and more governments are investing in the development of all kinds of linguistic resources that allow their citizens to be part of the new digital era, with all the benefits it entails. Language technology initiatives in the main regions of the world such as: Europe (Rehm et al., 2020; Nikulásdóttir et al., 2020; Meister et al., 2010; D'Halleweyn et al., 2006), India (Vikas, 2001; Choudhary, 2021), Africa (Grover et al., 2011), China (Kania et al., 2018), Saudi Arabia (Maegaard et al., 2008, 2005) and the Spanish speaking countries (Fernandez et al., 2016); allow us to attest how important language technologies have become in recent times.

In synchrony with all the developments mentioned above, it is time to talk about the efforts made for the development of the Faroese language in the digital sphere. The most recent initiative in this regard is the Ravnur Project, founded in the Faroe Islands. Thanks to the resources generated and shared by Ravnur, it has been possible to develop all the language resources presented in this document.

1.1 Faroese

The Faroe Islands is a set of small islands located at the North Atlantic in a half way between Scotland, Iceland and Norway. It is an autonomous territory of the Kingdom of Denmark with Faroese as the official language, which is spoken by around $54,000$ people. There are four main dialect areas in the Faroe Islands; north, northwest, central and southern (Petersen, 2022). The Faroe Islands is a bilingual country with Danish as the second official language. While many native speakers of Faroese use Danish for university education or employment in Denmark, Faroese is spoken as a first language by most of the population and is used on all domains, e.g. in education, public sectors, church etc. in the Faroe Islands. The first and, to this date, only Faroese speech synthesis was created in 2005 (Helgason and Gullbein, 2005) by combining efforts from researchers at the University of Stockholm and the University of the Faroe Islands and is used by the visually impaired community. Currently, there is a huge demand for Faroese ASR solutions, needed by the deaf, visually impaired and dyslexic communities - and also the general public, who wish to use their mother tongue when interacting with technology.

1.2 The Ravnur Project

The Faroese ASR research project, *Ravnur*, was assembled in 2019 (Foundation, 2019). The aim of the project was to create open-source resources that could be used to create automatic speech recognition (ASR) systems in Faroese. These resources would also be useful for creating other types of language technologies, as well as for lin-

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 32–41

guistic research. The project was funded by public and private initiators and investors, including the Faroese government. The development team consisted of a project leader, a technical leader, three native speaking junior linguists, an IT assistant, five university student assistants, as well as external advisors. The project concluded in the summer of 2022 with the publication of the Basic Language Resource Kit for Faroese (BLARK) (Simonsen et al., 2022; Debess et al., 2022).

1.3 Collection of the Speech Corpus

A Basic Language Resource Kit or BLARK is defined as the minimal set of language resources needed to create language and speech technology for a language (Krauwer, 2003; Maegaard et al., 2006). A BLARK is ideally language independent, but because languages may have different requirements, the contents of the BLARK may vary in some respects from language to language.

So, as Ravnur was an ASR project, the focus was on collecting good quality recordings of Faroese and creating a transcription corpus and pronunciation dictionary. During the course of the project, Ravnur collected 135 hours of recordings of 433 speakers total (249 female speakers and 184 male speakers) reading text of various genres, such as news, blogs, Wikipedia, law texts, GPS commands, word lists etc. The participants self-reported their gender, native language, dialect and age which varies between 15 to 83 years old. The recordings were made on TASCAM DR-40 Linear PCM audio recorders using the built-in stereo microphones in WAVE 16 bit with a sample rate of 48kHz. All recordings have been manually orthographically transcribed, while part of the speech corpus has been phonetically transcribed. The transcriptions were made by the university student assistants and the three Faroese linguists working for the project. All words that occur in the recordings were put in a pronunciation dictionary. The dictionary includes phonetic transcriptions written in SAMPA and PAROLE PoS-tags (Bilgram and Keson, 1998; Keson, 1998)[1].

As it can be seen, the BLARK developed by Ravnur is the starting point of the novel machine learning models presented in this work.

[1]Both the Faroese SAMPA alphabet (sometimes called FARSAMPA) and PAROLE PoS-tags were created by Ravnur for the BLARK.

2 The Ravnursson Corpus

Ravnursson[2] (Hernández Mena and Simonsen, 2022) is an ASR corpus with a length of 109 hours[3], extracted from the BLARK described in section 1.3. Unlike the original BLARK, the Ravnursson only contains the speech files along with their respective transcriptions. The main characteristics of the corpus are the following:

- The audio files in this corpus are distributed in a FLAC format at 16kHz@16bit mono.
- The corpus contains $71,949$ speech files from 433 speakers.
- The corpus is split into train, dev, and test portions. Lengths of every portion are: train = 100h08m, dev = 4h30m, test = 4h30m.
- The development and test portions have exactly 10 male and 10 female speakers each and both portions have exactly the same size in hours.
- As the test and development portions were selected to be gender balanced, an equal representation of all the dialectal variants is not guarantee in these two portions.
- Due to the limited number of prompts to read, only $39,945$ of the $71,949$ prompts in the whole corpus are unique. In other words, 44.48% of the prompts in the corpus are repeated at least once.
- Despite the repeated prompts in the corpus, the development and test portions do not share speakers with each other or with the training set.

2.1 Analysis of the Repeated Prompts

As the number of reading prompts for the corpus was limited during the recording process, the common denominator in the Ravnursson corpus is that one prompt is read by more than one speaker. This is relevant because it is a common practice in ASR

[2]As a matter of fact, the name Ravnursson comes from Ravnur (a tribute to the Ravnur Project) and the suffix "son" which in Icelandic means "son of". Therefore, the name "Ravnursson" means "The (Icelandic) son of Ravnur". The double "ss" is just for aesthetics.

[3]As it was mentioned in section 1.3, 135 hours of speech data were collected for the original BLARK. However, the Ravnursson Corpus contains 109 hours because we removed the portions with no presence of speech as much as we could.

to create a language model using the prompts that are found in the train portion of the corpus. That is not recommended for the Ravnursson Corpus as it counts with several prompts shared by all the portions and that will produce an important bias in the language modeling task.

Table 1 shows some statistics about the repeated prompts through all the portions of the corpus. The way this table has to be understood is as follows: for example, the first row indicates that there is a total of $71,949$ reading prompts in the whole corpus; $39,945$ of those are unique and $32,004$ are repeated at least once. Therefore, a total of 44.48% prompts in the whole corpus are repeated at least once. The same applies to the rest of the rows in Table 1.

Corpus Portion	Total Prompts	Unique Prompts	Repeat. Prompts	%
All	$71,949$	$39,945$	$32,004$	44.48%
Train	$65,616$	$38,646$	$26,970$	41.1%
Test	$3,002$	$2,887$	115	3.83%
Dev	$3,331$	$3,302$	29	0.87%

Table 1: Analysis of Repeated Prompts.

2.2 Corpus Organization

The "speech" directory contains all the speech files of the corpus. The files in the speech folder are divided in three directories: train, dev and test. The train portion is sub-divided in three types of recordings: RDATA1O, RDATA1OP and RDATA2; this is due the organization of the recordings in the original BLARK. There, the recordings are divided in Rdata1 and Rdata2.

One main difference between Rdata1 and Rdata2 is that the reading environment for Rdata2 was controlled by a software called "PushPrompt" which is included in the original BLARK (Simonsen et al., 2022). Another difference is that in Rdata1 there are some available transcriptions labelled at the phoneme level. The audio files in the speech directory of the Ravnursson corpus are divided in the folders RDATA1O where "O" is for "Orthographic" and RDATA1OP where "O" is for Orthographic and "P" is for phonetic. These categories are just a reminiscence of the original BLARK but it does not imply that the Ravnursson corpus comes with transcriptions at the phonetic level. In the case of the dev and test portions, the data come only from Rdata2 which does not have labels at the phonetic level in the original BLARK.

2.3 The Metadata File

The metadata file is a "tab-separated values file" (TSV) containing all the relevant information of the corpus. The file can be read using the Pandas (McKinney et al., 2010) library in Python and it comprises of the following 12 columns:

1. id: The filename without the extension ".flac".
2. speaker_id: The filename without the segment number.
3. filename: Full filename including the extension ".flac".
4. sentence_norm: The normalized transcription: no punctuation marks, no digits, lower case letters, one single space between words.
5. gender: The gender of the speaker: male or female.
6. age: The age range of the speaker: 15-35, 36-60, 61+ years old.
7. native_language: "Faroese" in all the cases.
8. dialect: The speaker dialect.
9. created_at: The date when the audio file was recorded.
10. duration: Duration of the speech file in seconds.
11. sample_rate: $16kHz$ in all the cases.
12. status: The corpus portion: train, test or dev.

2.4 Codification of the Audio Filenames

In the Ravnursson corpus, the filenames of the audio files encode relevant information about the respective speech files. The first row of Table 2, shows a typical audio filename. The second row enumerates the fields of information encoded in the filename and the third row shows the same filename of row one but broken down in the eight parts as specified in the second row.

The explanation of the information encoded in the filename is at follows:

1. Gender of the Speaker: **M** for male or **K** for female

$MEY01\_040319\_rok0\_0009.flac$							
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
M	E	Y	01	040319	rok0	0009	.flac

Table 2: Audio Filename Format.

2. Dialect Group: **U** for Suðuroy, **A** for Sandoy, **S** for Suðurstreymoy, **E** for Norðurstreymoy/Eysturoy (exclusive of Eiði, Gjógv og Funningur), **V** for Vágar and **N** for Norðuroyggjar (inclusive of Eiði, Gjógv og Funningur)

3. Age Group: **Y** for "Younger" between 15-35 years old, **M** for "Middle-aged" between 36-60 years old and **E** for "Elderly" 61 years old or older.

4. Number of Speaker in a Group: is a number that always consists of two digits and starts with 01, 02, 03 etc. The first speaker in a group with the same gender, dialect group and age group (e.g. MEY) gets the number 01. The next speaker in the same group gets the number 02 (and his ID is therefore MEY02).

5. Date: The date when the speech was recorded (day/month/year).

6. Type of reading material: This code can only be found in speech files at RDATA1O and RDATA1OP. For more information about the types of reading material please see the documentation of the original BLARK and its directory "readingtexts_1.0".

7. Segment Number: In the original BLARK the recording session is distributed as one audio file per speaker and it can be very long from the ASR perspective. So, the audio files are subdivided in segments of around 10 seconds to fit most of the modern ASR engines[4]. The numbering is continuous for each speaker; the only exception is with the files MUY01_180519_set4_0004 and MUY02_190120_eind2_0007. We detected that they are empty and we removed them.

8. File extension: The corpus is distributed in FLAC format.

3 Acoustic Models

The development of the Ravnursson corpus allowed us to create acoustic models in four different ASR systems: WAV2VEC2, NeMo, Kaldi and PocketSphinx. In this section we discuss the details of how we created each of them.

3.1 WAV2VEC2 Model

WAV2VEC, released in 2019, is a convolutional neural network that takes raw audio as input and computes a general representation that can be input to a speech recognition system (Schneider et al., 2019). In 2020, a second version, WAV2VEC2 (Baevski et al., 2020) was released. Based on WAV2VEC2, the XLSR-53 (Conneau et al., 2020) was also released in 2020. XLSR-53 is a open-source model trained with more than 50k hours of unlabelled speech in 53 languages. It can be used to create acoustic models in any language through a fine-tuning step.

Using the XLSR-53 as a starting point, we created an acoustic model suitable for Faroese (Hernandez Mena, 2022b)[5] which is available on a Creative Commons licence "CC BY 4.0". The fine-tuning process for this model lasted 30 epochs. Due to the acceptable WER results that we obtained with this model, we decided not to add any type of augmentation to the training data.

3.2 NeMo Model

NeMo (Neural Modules) is a Python toolkit developed by NVIDIA for creating AI applications. It comes with extendable collections of pre-built modules for automatic speech recognition and natural language processing (Kuchaiev et al., 2019). One of the NeMo modules suitable for speech recognition is called Quartznet (Kriman et al., 2020) which is a convolutional model trained with Connectionist Temporal Classification (Graves, 2012) or CTC for short.

In order to train an ASR model for Faroese in NeMo, we used the public checkpoint

[4]According to the developers of Sphinx, the optimal length for audio recordings in ASR is between 5 and 30 seconds (see `https://cmusphinx.github.io/wiki/tutorialam/`. However, we segmented the audio files of the Ravnursson Corpus to have a lenght around 10 seconds to fit the format of other corpora developed by our laboratory

[5]Available at: `https://huggingface.co/carlosdanielhernandezmena/wav2vec2-large-xlsr-53-faroese-100h`

	Points of articulation									
Manners of articulation	**Consonants**	**Bi-labial**	**Labiodental**	**Dental**	**Alveolar**	**Post-alveolar**	**Retroflex**	**Palatal**	**Velar**	**Glottal**
	Voiceless Stop	p			t				k	
	Voiced Stop	b			d				g	
	Voiceless Affricate					tS				
	Voiced Affricate					dZ				
	Voiceless Fricative		f	5	s	S	z			h
	Voiced Fricative		v	4						
	Voiceless Nasal	M			x				X	
	Voiced Nasal	m			n				N	
	Voiceless Lateral				L					
	Voiced Lateral				l					
	Approximants				r			j	w	
	Vowels					**Front**		**Central**		**Back**
	Close					i y		3		u
							I Y		U	
	Close-mid					e	2			o
								8		
	Open-mid						E 9			O
	Open						a			

Table 3: Phonetic Repertoire of Faroese

"QuartzNet15x5Base-En.nemo[6]" as a starting point. This model was trained with more than 3k hours of English data in a Quartznet architecture during 600 epochs. Based on a work by Huang et al.[7], we fine-tuned the checkpoint with the data of the Ravnursson corpus during 236 epochs, obtaining a first checkpoint able to recognize Faroese. Then, we augmented the initial 100 hours of the training portion of the Ravnursson corpus to 300 hours through speech perturbation using two speed rates: 0.9 and 1.1. Finally, we fine-tuned our initial checkpoint in Faroese with the augmented data during 163 epochs to obtain a final model[8] (Hernandez Mena, 2022a) which is available on a Creative Commons licence "CC BY 4.0".

3.3 Kaldi Model

Kaldi (Povey et al., 2011), released in 2011, is a well established toolkit for speech recognition written in C++, which is based on distinct paradigms such as: finite-state transducers (Allauzen et al., 2007), Hidden Markov Models (Juang and Rabiner, 1991), Gaussian Mixture Models (Naeem et al., 2020) as well as neural networks (Rath et al., 2013).

Our "Kaldi Recipe for Faroese[9]" (Hernández Mena, 2022) was created using the Ravnursson corpus as training data. The recipe produces models based on Hidden Markov Models (HMMs) as well as Neural Networks; in specific, the neural network is an LSTM or "Long Short-Term Memory" (Huang et al., 2017) and it uses speed perturbation as augmentation technique with speed rates of 0.9 and 1.1. This recipe requires a 3-gram language model (lm) for decoding, a 4-gram lm for re-scoring and a pronouncing dictionary; elements that are available in our "Faroese Language Models with Pronunciations" (Hernández Mena et al., 2022), discussed in further sections.

[6]Available at: https://catalog.ngc.nvidia.com/orgs/nvidia/models/nemospeechmodels/files

[7]The decision of using the QuartzNet architecture and not others, was based mainly on this research paper. A comparison of different NeMo architectures is beyond the scope of this paper.

[8]Available at: https://huggingface.co/carlosdanielhernandezmena/stt_fo_quartznet15x5_sp_ep163_100h

[9]See: https://github.com/CarlosDanielMena/Kaldi_Recipe_for_Faroese

The recipe is available on Clarin.is [10] under a Creative Commons licence "CC BY 4.0".

3.4 PocketSphinx Model

Sphinx is an old speech recognition system based on Hidden Markov Models developed by Carnegie-Mellon University in the late 80's (Lee et al., 1990). Through time, progressive versions of Sphinx have been released up the version 4. At some point, the version 2 turned into PocketSphinx (Huggins-Daines et al., 2006). PocketSphinx was supposed to be a lighter and faster version of Sphinx but nowadays it has become the main version that can be used in real time mode, even in ARM processors. PocketSphinx has long ceased to be a suitable system for research, but nevertheless it still has an active community of users that choose it as a real time speech recognition system in devices with not a great computing power such as Raspberry PI (Upton and Halfacree, 2014) or other ARM computers.

Our PocketSphinx models[11], trained with the Ravnursson corpus, are suitable for the PocketSphinx Python library available at the Pypi repository [12]. With this library it is possible to perform both standard and real time speech recognition, forced-alignment and produce timestamps. The version of PocketSphinx that was available when we produced these models was the number 4. Few weeks later the version 5 was released but our models remain compatible.

The example language model that comes with the PocketSphinx model is a 3-gram model created using the training prompts of the Ravnursson Corpus. The test portion of the corpus was used to measure a WER of 18.7%. We don't show this result in Table 5 because the use of the training prompts in the language model produces a bias that is not fair to the other models as we point out in section 2.1. We strongly recommend to create a language model for the specific task that is required and to kept it as short as possible because a larger model will impact the latency of the system.

[10] See: http://hdl.handle.net/20.500.12537/305

[11] Available at: https://github.com/CarlosDanielMena/RAVNURSSON_FAROESE_Models_100h

[12] See: https://pypi.org/project/pocketsphinx/

4 Pronunciation Models

The pronunciation models that we discuss in this section is a set of pronouncing dictionaries that are included in our "Faroese Language Models with Pronunciations" (Hernández Mena et al., 2022) along with a number of language models that will be discussed in section 5. Most of the pronunciations come from the original BLARK, but for convenience, we subdivide them in different dictionaries as follows:

- Central_Faroese.dic: It contains pronunciations of the variant of Faroese which is spoken in the capital.
- East_Faroese.dic: It contains pronunciation of the northwest variant of Faroese[13].
- Ravnursson_Composite_Words.dic: It contains words with hyphens and/or underscores that are present in the Ravnursson Corpus. We keep them separate in a different dictionary because these type of composite words can be problematic for a grapheme-to-phoneme (g2p) tool.
- BLARK.dic: It contains pronunciations of words that are present in the BLARK but that are not present in any other dictionary of the set.
- FAROESE_ASR.dic: This dictionary is recommended for ASR experiments in Kaldi or any other ASR system based on phonemes. The dictionary is the mix of Central_Faroese.dic, East_Faroese.dic and Ravnursson_Composite_Words.dic. It is important to clarify that the dictionary can contain words with multiple pronunciations, which is normal in Kaldi-like systems.

4.1 Phoneme Sets of Dictionaries

Table 3 shows the phonetic repertoire of Faroese using 42 SAMPA symbols. Each of these correspond to an individual phoneme that is included

[13] In the most recent dialect classification (Petersen, 2022), the islands in the northwest area are classified as being the same dialect area. However, there is a difference in the pronunciation of the digraph *ei* between the westernmost islands and the more central and eastern islands in that dialect area. Therefore, the westernmost part of the dialect area is not included in our EAST dictionary. For that reason, we have given this dictionary the name EAST. The idea is that this makes it is possible to make WEST, NORTH and SOUTH dictionaries in the future.

SAMPA	IPA	SAMPA	IPA	SAMPA	IPA	SAMPA	IPA
p	p^h	m	m	e	e	aJ	ai
b	b	M	m̥	E	ɛ	aW	au
t	t^h	n	n	a	a	OJ	ɔi
d	d	x	n̥	y	y	OW	ɔu
k	k^h	N	ŋ	Y	ʏ	3W	ʉu
g	g	X	ŋ̊	2	ø	EW	eu
f	f	l	l	9	œ	9W	œu
v	v	L	l̥	u	u	9J	œi
s	s	j	j̊	o	o	4	ð
S	ʃ	w	w	O	ɔ	5	θ
z	ʂ	r	ɹ	EA	ɛa	8	ə
h	h	U	ʊ	OA	ɔa	H	Pre-aspiration
tS	$tʃ^h$	i	i	UJ	ʊi		
dZ	ʤ	I	ɪ	EJ	ɛi		

Table 4: SAMPA vs. IPA Equivalences.

in the pronouncing dictionaries described in section 4, except for the vowel "/3/" that only occurs in diphthong. The phonetic repertoire of Faroese includes the following 12 diphthongs: **EA**, **OA**, **UJ**, **EJ**, **aJ**, **aW**, **OJ**, **OW**, **3W**, **EW**, **9W** and **9J**. Summing the 41 individual phonemes in Table 3, plus the 12 diphthong, plus seven phonemes with pre-aspiration (**Hb**, **Hd**, **HdZ**, **Hg**, **Hp**, **Ht**, **HtS**), we have a total of 60 phonemes. That is the list of 60 phonemes that are included in the dictionaries presented in section 4. To see an equivalence between our SAMPA symbols versus the IPA phonemes, please see Table 4.

5 Language Models

As it was mentioned in section 4, our "Faroese Language Models with Pronunciations" is a set of n-gram language models of distinct sizes that were created using the Faroese text provided in the BLARK, as it provides with text from newspaper articles, parliamentary speeches, books and more. The normalization process of that text included to change everything to lowercase, allow only characters belonging to the Faroese alphabet and removing punctuation marks.

The resulting text has a length of more than half million lines of text (106.3MB approximately). The text was used to create a 3-gram (recommended for decoding) and a 4-gram (recommended for re-scoring) language models with the SRILM toolkit (Stolcke, 2002). Both the 3-gram and 4-gram models come in pruned and unpruned versions. It also includes a 6-gram language model in binary format suitable for ASR experiments with the NeMo toolkit. In particular, this model was created using KenLM (Heafield, 2011). It is important to mention that all the words present in any of the language models are present in the pronouncing dictionaries for the east and central variants of Faroese (see section 4).

6 Results

Table 5 shows a comparison of the Word Error Rate (WER) obtained with the acoustic models presented in section 3 with the exception of the PocketSphinx models as discussed in section 3.4.

The NeMo results include the WER obtained using the 6-gram language model (LM) presented in section 5 as well as the WER obtained with no language model at all. The Kaldi results include the WER obtained with Hidden Markov Models (HMM) only and the WER obtained with the LSTM network. As it can be seen, the best results are obtained with the WAV2VEC2 model, which is not a surprise as it is well known that it can achieve acceptable results with less than 1 hour of speech data. What is remarkable indeed, is the gap of performance between WAV2VEC2 and the other systems.

In addition to this, based on our previous experience (Hernandez Mena et al., 2020; Mena et al., 2022), it is also remarkable that the WER obtained with NeMo using a language model and the WER obtained with Kaldi using the LSTM are so close to each other despite of the relatively low amount of training data. This fact reveals that the train-

Corpus Portion	NeMo SP No LM	NeMo SP With LM	Kaldi HMM	Kaldi LSTM	WAV2VEC2 XLRS-53
Dev	20.51%	13.66%	20.60%	12.22%	**5.56%**
Test	22.81%	15.95%	23.44%	14.04%	**7.60%**

Table 5: WER Results.

ing method described by Huang et al. in 2020 and the use of speed perturbation for training are really effective in NeMo.

On the other hand, Table 6 shows the results obtained with the newest system Whisper (Radford et al., 2022). Whisper is a transformer-based speech recognition system trained with 680k hours of transcribed data in multiple languages. Whisper is also a multitask system able to perform multilingual speech recognition as well as speech translation and language identification. According to the original paper (Radford et al., 2022), the training set that Whisper uses for translation includes 46 hours of Faroese. Based on this, we decided to test Whisper in its distinct sizes with no fine-tuning step and using the development and test portions of the Ravnursson corpus. As it can be seen in Table 6, we obtained terribly bad WER results, revealing that Whisper needs to be fine-tuned prior to recognize Faroese data; unfortunately, this is beyond the scope of this paper but it will tackle as further work.

Whisper Size	Dev WER	Test WER
Tiny	113.4%	116.7%
Base	112.61%	113.07%
Small	128.05%	132.64%
Medium	116.34%	119.3%
Large	105.93%	110.25%

Table 6: Whisper WER Results.

7 Conclusions

A major development of Faroese ASR is presented in this work. The Ravnursson project has produced a corpus of 109 hours of transcribed speech and acoustic models for WAV2VEC2, NeMo, Kaldi and PocketSphinx have been developed. Furthermore, the project has also produced a set of n-gram language models of distinct sizes and pronunciation dictionaries in Faroese suitable for ASR experimentation. Quality assessment of the acoustic models are shown in Table 5 where the best results of 7.60% WER was achieved by the WAV2VEC2 model. Another interesting result is shown in Table 6 demonstrating that a fine-tuning step is needed for Faroese for the multi-lingual ASR system Whisper.

Faroese ASR is no longer under-developed due to this work. The project has lowered the technological threshold for implementing ASR solutions for Faroese in industry and for studying the Faroese language using ASR as a tool. With all the results made available with open licenses, there is no good reason why Faroese ASR should not be included in standard language technology software in the future.

8 Further Work

As further work, it is clear to us that we have to explore acoustic models with the new parameter versions of WAV2VEC2 such as 300m, 1B and 2B; as well as the Whisper system with a fine-tuning step in Faroese in order to keep improving our WER results. Another future challenge is to add more Faroese data to our models, including conversational speech.

Acknowledgments

This project was made possible under the umbrella of the Language Technology Programme for Icelandic 2019-2023. The Programme, which is managed and coordinated by Almannarómur, is funded by the Icelandic Ministry of Education, Science and Culture.

Special thanks to the Ravnur Project for making their "Basic Language Resource Kit" (BLARK 1.0) publicly available.

References

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. Openfst: A general and efficient weighted finite-state transducer library. In *International Conference on Implementation and Application of Automata*, pages 11–23. Springer.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Thomas Bilgram and Britt Keson. 1998. The construction of a tagged danish corpus. In *Proceedings of the 11th Nordic Conference of Computational Linguistics (NODALIDA 1998)*, pages 129–139.

Narayan Choudhary. 2021. Ldc-il: The indian repository of resources for language technology. *Language Resources and Evaluation*, 55(3):855–867.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2020. Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*.

Iben Nyholm Debess, Sandra Saxov Lamhauge, Annika Simonsen, Peter Juel Henrichsen, Egil Hofgaard, Uni Johannesen, Petur Markus Josenius Hammer, Gunnvør Hoydal Brimnes, Ebba Malena Debess Thomsen, and Beinta Poulsen. 2022. Basic language resource kit 1.0 for faroese. *OpenSLR.org*.

Elisabeth D'Halleweyn, Jan Odijk, Lisanne Teunissen, and Catia Cucchiarini. 2006. The dutch-flemish hlt programme stevin: Essential speech and language technology resources. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*.

David Pérez Fernandez, Doaa Samy, and Juan de Dios Llorens Gonzalez. 2016. Spanish language technologies plan. In *International Workshop on Future and Emerging Trends in Language Technology*, pages 50–60. Springer.

Talutøkni Foundation. 2019. The project ravnur. In *Talutøkni Foundation*.

Alex Graves. 2012. Connectionist temporal classification. In *Supervised sequence labelling with recurrent neural networks*, pages 61–93. Springer.

Aditi Sharma Grover, Gerhard B Van Huyssteen, and Marthinus W Pretorius. 2011. The south african human language technology audit. *Language resources and evaluation*, 45(3):271–288.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Pétur Helgason and Sjúrður Gullbein. 2005. Færøsk talesyntese: Rapport marts 2005. *Nordisk sprogteknologi 2005-Nordic Language Technology*, page 51.

Carlos Daniel Hernandez Mena. 2022a. Acoustic model in faroese: stt_fo_quartznet15x5_sp_ep163_100h. *huggingface.co*.

Carlos Daniel Hernandez Mena. 2022b. Acoustic model in faroese: wav2vec2-large-xlsr-53-faroese-100h. *huggingface.co*.

Carlos Daniel Hernández Mena. 2022. Kaldi recipe for faroese. *Clarin.is*.

Carlos Daniel Hernandez Mena, Albert Gatt, Andrea DeMarco, Claudia Borg, Lonneke van der Plas, Amanda Muscat, and Ian Padovani. 2020. Masri-headset: A maltese corpus for speech recognition. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6381–6388, Marseille, France. European Language Resources Association.

Carlos Daniel Hernández Mena, Sandra Saxov Lamhauge, Iben Nyholm Debess, and Annika Simonsen. 2022. Faroese language models with pronunciations. *Clarin.is*.

Carlos Daniel Hernández Mena and Annika Simonsen. 2022. Ravnursson faroese speech and transcripts. *Clarin.is*.

Jocelyn Huang, Oleksii Kuchaiev, Patrick O'Neill, Vitaly Lavrukhin, Jason Li, Adriana Flores, Georg Kucsko, and Boris Ginsburg. 2020. Cross-language transfer learning, continuous learning, and domain adaptation for end-to-end automatic speech recognition. *arXiv preprint arXiv:2005.04290*.

Lu Huang, Ji Xu, Jiasong Sun, and Yi Yang. 2017. An improved residual lstm architecture for acoustic modeling. In *2017 2nd International Conference on Computer and Communication Systems (ICCCS)*, pages 101–105. IEEE.

David Huggins-Daines, Mohit Kumar, Arthur Chan, Alan W Black, Mosur Ravishankar, and Alexander I Rudnicky. 2006. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *2006 IEEE international conference on acoustics speech and signal processing proceedings*, volume 1, pages I–I. IEEE.

Biing Hwang Juang and Laurence R Rabiner. 1991. Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272.

Elsa Kania, Paul Triolo, and Graham Webster. 2018. Translation: Chinese government outlines ai ambitions through 2020. *New America*.

Britt Keson. 1998. Vejledning til det danske morfosyntaktisk taggede parole-korpus. *Parole report, Det Danske Sprog-og Litteraturselskab (DSL)*.

Steven Krauwer. 2003. The basic language resource kit (blark) as the first milestone for the language resources roadmap. In *Proceedings of SPECOM*, page 15.

Samuel Kriman, Stanislav Beliaev, Boris Ginsburg, Jocelyn Huang, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, and Yang Zhang. 2020. Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6124–6128. IEEE.

Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, et al. 2019. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577*.

K-F Lee, H-W Hon, and Raj Reddy. 1990. An overview of the sphinx speech recognition system. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(1):35–45.

Bente Maegaard, Mohammed Atiyya, Khalid Choukri, Steven Krauwer, Chafic Mokbel, and Mustafa Yaseen. 2008. Medar: Collaboration between european and mediterranean arabic partners to support the development of language technology for arabic. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Bente Maegaard, Khalid Choukri, Chafik Mokbel, and Mustafa Yaseen. 2005. *Language technology for Arabic*. NEMLAR, Center for Sprogteknologi, University of Copenhagen.

Bente Maegaard, Steven Krauwer, Khalid Choukri, and Lise Damsgaard Jørgensen. 2006. The blark concept and blark for arabic. In *LREC*, pages 773–778.

Wes McKinney et al. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, pages 51–56. Austin, TX.

Einar Meister, Jaak Vilo, and Neeme Kahusk. 2010. National programme for estonian language technology: a pre-final summary. In *Human Language Technologies–The Baltic Perspective*, pages 11–14. IOS Press.

Carlos Daniel Hernandez Mena, David Erik Mollberg, Michal Borskỳ, and Jón Guðnason. 2022. Samrómur children: An icelandic speech corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 995–1002.

Saad Naeem, Majid Iqbal, Muhammad Saqib, Muhammad Saad, Muhammad Soban Raza, Zaid Ali, Naveed Akhtar, Mirza Omer Beg, Waseem Shahzad, and Muhhamad Umair Arshad. 2020. Subspace gaussian mixture model for continuous urdu speech recognition using kaldi. In *2020 14th International Conference on Open Source Systems and Technologies (ICOSST)*, pages 1–7. IEEE.

Anna Björk Nikulásdóttir, Jón Guðnason, Anton Karl Ingason, Hrafn Loftsson, Eiríkur Rögnvaldsson, Einar Freyr Sigurðsson, and Steinthór Steingrímsson. 2020. Language technology programme for icelandic 2019-2023. *arXiv preprint arXiv:2003.09244*.

Hjalmar P Petersen. 2022. Evidence for the modification of dialect classification of modern spoken faroese. *European Journal of Scandinavian Studies*, 52(1):43–58.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, CONF. IEEE Signal Processing Society.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.

Shakti P Rath, Daniel Povey, Karel Veselỳ, and Jan Cernockỳ. 2013. Improved feature processing for deep neural networks. In *Interspeech*, pages 109–113.

Georg Rehm, Katrin Marheinecke, Stefanie Hegele, Stelios Piperidis, Kalina Bontcheva, Jan Hajič, Khalid Choukri, Andrejs Vasiljevs, Gerhard Backfried, Christoph Prinz, et al. 2020. The european language technology landscape in 2020: Language-centric and human-centric ai for cross-cultural communication in multilingual europe. *arXiv preprint arXiv:2003.13833*.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Annika Simonsen, Sandra Saxov Lamhauge, Iben Nyholm Debess, and Peter Juel Henrichsen. 2022. Creating a basic language resource kit for faroese. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4637–4643.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*.

Eben Upton and Gareth Halfacree. 2014. *Raspberry Pi user guide*. John Wiley & Sons.

Om Vikas. 2001. Language technology development in india. *Ministry of Information Technology*.

Good Reads and Easy Novels
Readability and Literary Quality in a Corpus of US-published Fiction

Yuri Bizzoni
Center for Humanities Computing
Aarhus University, Denmark
yuri.bizzoni@cc.au.dk

Pascale Feldkamp Moreira
School of Communication and Culture
Aarhus University, Denmark
pascale.moreira@cc.au.dk

Nicole Dwenger
Aarhus, University, Denmark
01805351@post.au.dk

Ida Marie S. Lassen
Center for Humanities Computing
Aarhus University, Denmark
idamarie@cas.au.dk

Kristoffer L. Nielbo
Center for Humanities Computing
Aarhus University, Denmark
kln@cas.au.dk

Mads Rosendahl Thomsen
School of Communication and Culture
Aarhus University, Denmark
madsrt@cc.au.dk

Abstract

In this paper, we explore the extent to which readability contributes to the perception of literary quality as defined by two categories of variables: expert-based (e.g., Pulitzer Prize, National Book Award) and crowd-based (e.g., GoodReads, WorldCat). Based on a large corpus of modern and contemporary fiction in English, we examine the correlation of a text's readability with its perceived literary quality, also assessing readability measures against simpler stylometric features. Our results show that readability generally correlates with popularity as measured through open platforms such as GoodReads and WorldCat but has an inverse relation with three prestigious literary awards. This points to a distinction between crowd- and expert-based judgments of literary style, as well as to a discrimination between fame and appreciation in the reception of a book.

1 Introduction and Related Works

Is it overall better for a novel to strive for an easy prose, or is there a link between difficulty and literary quality? The concept of readability has been studied for decades and is defined as the ease with which a text can be read and understood (Dale and Chall, 1949). Several works have attempted to define an easy way to compute readability in order to make, for example, didactic books more accessible, reduce technical jargon in documents produced for the general public, and adjust text selections according to the intended audience (Dubay, 2004). The result has been a series of popular and amply tested measures, each with a slight difference in their model of readability. Dale and Chall (1949), for example, referred to readability as the combination of elements in a text that impact important aspects of a reader's experience - including whether the reader can understand the text, finds it interesting, and can read with optimal speed (Dale and Chall, 1949). Despite their shortcomings (Redish, 2000), readability measures have been broadly applied to a large number of different domains. Measures of readability vary according to what aspect of a text they take into account, but they typically combine features such as sentence length, word length, and the presence of complex words. While the actual ease of a text depends on reader characteristics (background, situation, ability) it is widely accepted that simple textual features such as sentence length, syllables per word and lexical diversity impact the reading experience (Dubay, 2004). The connection of readability to the quality of a text has often been implied when it comes to non-fiction, and early studies into readability attest to the educational and social importance of developing such measures to improve technical or expository documents (Chall, 1947), but its role in the quality of *literary* fiction is much more complex. An easy-to-read novel can be enjoyable to read, but may also apppear poor or uno-

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 42–51

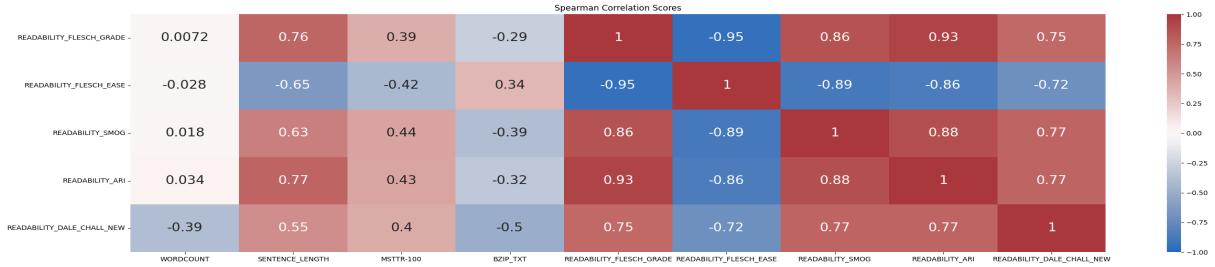


Figure 1: Correlations between stylometrics and flavours of readability (Spearman). All correlations between 0.09 and 0.99 are statistically significant.

riginal. In literary studies, the idea that readability might be a precondition for literary success is debated, and literary texts have been assessed variously by readability measures and similar metrics. Sherman (1893) was one of the first scholars to propose certain values of average sentence-length and reading ease as properties of "better" literary style. Readability naturally varies across genre, but it is a widespread conception for readers and publishers alike that bestsellers (as defined by top book-sales) are easier to read (Martin, 1996). More recently, readability has gained traction in areas of (commercial) creative writing and publishing, especially where its measures are implemented in text-editing tools such as the Hemingway or Marlowe editors [1]. These applications tend to favour lower readability scores - which is, texts easier to read. Yet, on the large scale, few studies have included readability as a measure that could help predicting literary quality. Studying a small corpus of bestsellers and more literary, canonical works, Martin (1996) found no significant difference in readability, using a modified Flesch reading score, while Garthwaite (2014) found differences in readability between bestsellers and commercially endorsed book-list titles. Relying on multiple measures of readability and one measure of literary quality (i.e., GoodReads' average ratings), Maharjan et al. (2017) found that readability was actually a weak measure for estimating popularity in comparison to, for example, character n-grams. Still, many studies of literary success, popularity, or perceived literary quality have sought to approximate text complexity and have studied textual properties upon which formulae of readability are directly or indirectly based, such as sentence-length, vocabulary richness, or text compressibility (Brottrager et al., 2022; van Cranenburgh and Bod, 2017; Crosbie et al., 2013).

The question of the role of readability in literary quality is complicated by the practical and conceptual problem of defining literary quality itself, and consequently of quantifying it for large scale studies. Studies that seek to predict perceived literary quality from textual features often rely on the provisional proxy of one single gold standard, such as book-ratings from large user-platforms like GoodReads (Maharjan et al., 2018), personally or institutionally compiled canons (Mohseni et al., 2022) or sales-numbers (Wang et al., 2019). However, it has been shown that readers may have different, distinct perceptions of quality that are not necessarily based on the same criteria or prompted by the same textual features (Koolen et al., 2020).

In this paper, we explore to what extent readability might contribute to the perception of literary quality – defined through several alternative measures – in a large fiction corpus of modern and contemporary novels in English, taking into account, instead of one golden standard, different contextual perspectives on literary quality, so as to cover both crowd-based and "expert"-based standards of judgment.

2 Data and Methods

The essence of our approach consists in examining whether readability, as measured through five different algorithms, and literary quality, as approximated through six different resources, show any correlation on a large corpus of English-language fiction. We use standard correlation measures (Pearson and Spearman product-moment correlation coefficients r_p and r_s, respectively). For inference on the correlation measures, simple Student's t-tests are used. For robustness checks, correlation coefficients were also modelled using a Bayesian ridge model of standardized the variables – al-

[1] https://hemingwayapp.com/help.html, https://authors.ai/marlowe/

though not reported due to limited space.[2]

2.1 Corpus

We use a corpus of modern and contemporary fiction in English, the so-called Chicago Corpus. [3] The Chicago Corpus is a collection of over 9000 novels from 1880 to 2000, representing works of fiction that are widespread in libraries, that is, the works of fiction that have a large number of library holdings as listed on WorldCat, a large-scale, international online library catalogue [4]. The number of holdings was used as a first filtering measure to include or exclude works in the dataset, yet there are still large differences in how many libraries hold each title, so we can use it as a metric to score different titles within the dataset as well. The corpus is unique, to our knowledge, for its diversity and extraordinary representation of famous popular- and genre-fiction, as well as seminal works from the whole period: key works of modernism and postmodernism as well as Nobel laureates and winners of major literary award. Still, it should be noted that the Chicago corpus reflects a clear cultural and geographical tilt, with a strong over-representation of Anglophone authors, and features only works either written in or translated into English. This tilt should be taken into account especially since we correlate textual features in the corpus to readability measures that were developed - and are particularly successful - in the English language context (Antunes and Lopes, 2019).

	N. Titles	N. Authors
Whole corpus	9089	7000
Pulitzer	53	46
NBA	104	79
Hugo	96	47

Table 1: Overall titles and authors in the corpus and number of long-listed titles for each award.

2.2 Measures of quality

We use six different measures of literary quality of two main types, heuristically setting up a qualitative distinction between more crowd-based and more expert-based measures. Expert-based measures may be supposed more institutionally prescribed, where titles are distinguished by appointing committees (as with literary prizes). Here, we chose to look at three prominent literary prizes in Anglophone literary culture: The Pulitzer Prize, the National Book Award, and the Hugo Awards, considering titles that were both long- and short-listed for these prizes. The selection of awards allows us to consider a main-stream vs. genre-literature divide in our expert measures, since the first two prizes are assigned mainly to works of literary fiction, while the latter is an award given to works of genre fiction (science fiction and fantasy).

Crowd-based measures may be considered more democratic in the sense of being user-created, for example by users' ratings on large scale reading community sites such as GoodReads, or by the effect of popular demand on library acquisitions. We use three standards here: the average ratings of titles on GoodReads (from 0 to 5 stars), the average rating count of titles on GoodReads (number of ratings given to a given title), and the number of libraries that hold a title according to Worldcat. Goodreads ratings and/or rating counts are often favoured in studies of literary quality and reception, because they seem to proffer more democratic literary evaluations "in the wild", considering the large diversity and geographical spread of its nearly 90 million users (Nakamura, 2013). In slight contrast to Goodread's ratings, we consider library holdings a conceptually hybrid measure, standing between completely free reader-based votes and expert-driven choices, as libraries respond to user-demand from within an institutional structure.

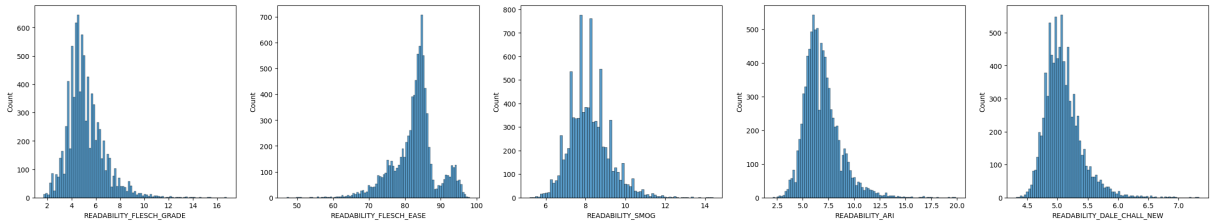
2.3 Measures of readability

For assessing the complexity and/or difficulty of literary texts, we apply various measures of readability. Since the 1920s, and especially with the success of the Flesch and Dale-Chall formulas in the 1950s, combinations of sentence-length and words and/or syllables have been used to assess the difficulty of a text as proxies of word and sentence complexity (Dale and Chall, 1948). According to Dubay (2004), there were more than 200 different versions of readability formulas in 1980, while new ones are still introduced and old ones revised. Still, measures from what Dubay calls the "classic" readability studies, continue to be the

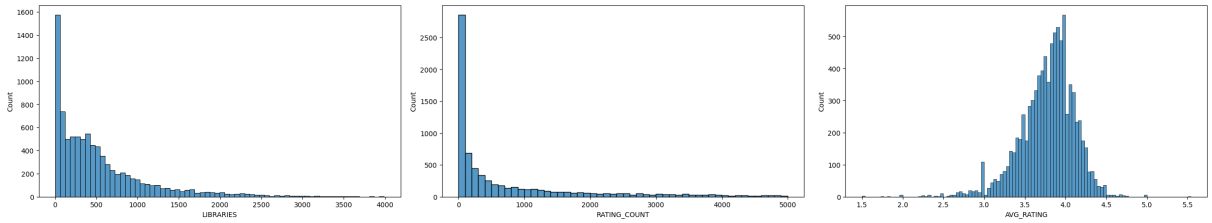
[2]The code will be publicly available upon acceptance.

[3]While we cannot directly provide access to the corpus, it is possible to contact the authors for requests.

[4]https://www.worldcat.org/about

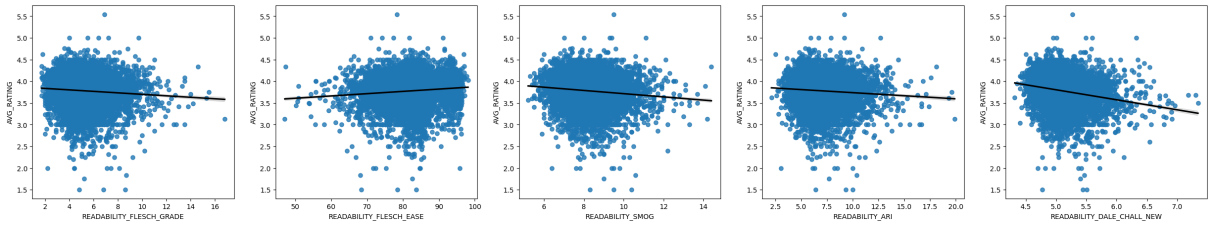


(a) Distributions of readability measures

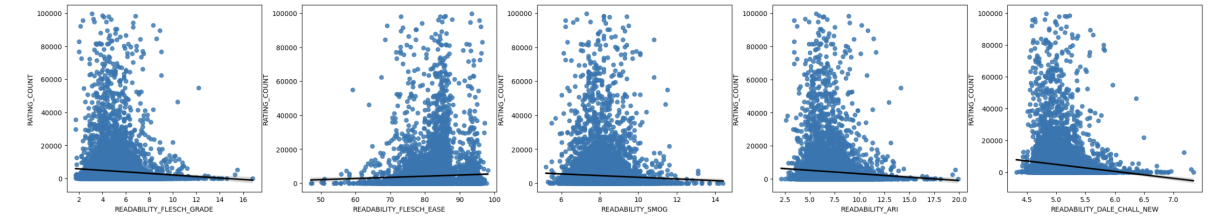


(b) Distributions of quality measures. Rating count is visualised with cutoff at 5000 for legibility.

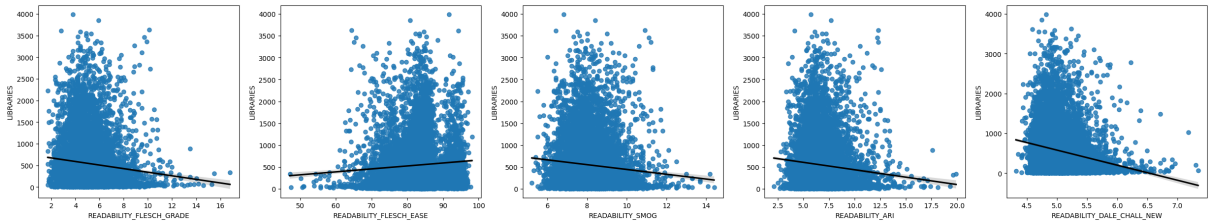
Figure 2: Distributions of measures



(a) Readability and GoodReads' average rating



(b) Readability and GoodReads' rating count, with cutoff at 100.000



(c) Readability and library holdings

Figure 3: Quality standards and flavours of readability

most widely used measures and to prove themselves effective in assessing text difficulty (Dubay, 2004; Stajner et al., 2012) - despite their relative simplicity (being counts of two or three aspects of texts). As mentioned, readability is subjective and depends on the audience/reader. However, if the intended audience or specific reader is unkown (as in our case), readability scores may provide a general/overarching measure which is also sufficient for comparison between texts. These measures have been applied to a wide range of written productions, from technical and journalistic texts to fiction. Flesch, for example, found that fiction tend to score a *Flesch Reading Ease* score in the

range 70 ¡ Score ¡ 90, in contrast to scientific text that often score below 30 (Flesch, 1948). In the present study we used five different "classic" readability algorithms to measure the prose of each book, chosen for their popularity and interpretability [5].

- The *Flesch Reading Ease* is a measure of readability based on the average sentence length (ASL), and the average syllables per word (word length)(ASW). It is calculated as follows:

$$\text{Score} = 206.835 - (1.015 \times \text{ASL}) - (84.6 \times \text{ASW})$$

- The *Flesch-Kincaid Grade Level* is a revised version of the Flesch Reading Ease score. Like the former, it is based on the average sentence length (ASL), and the number of syllables per word (ASW). It is calculated as follows:

$$\text{GL} = (0.4 \times \text{ASL}) + (12 \times \text{ASW}) - 15$$

- The *SMOG Readability Formula* is a readability score introduced by McLaughlin (McLaughlin, 1969). It measures readability based on the average sentence length and number of words with more than 3 syllables (number of polysyllables), applying the formula:

$$\text{SMOG grading} = 3 + \sqrt{polysyllablecount}$$

- The *Automated Readability Index* is a readability score based on the average sentence length and number of characters per words (word length). It is calculated as follows:

$$4.71\frac{\text{characters}}{\text{words}} + 0.5\frac{\text{words}}{\text{sentences}} - 21.43$$

- The *New Dale–Chall Readability Formula* is a 1995 revision of the Dale-Chall readability score (Chall and Dale, 1995). It is based on the average sentence length (ASL) and the percentage of "difficult words" (PDW) which were defined as words which do not appear on a list of words which 80 percent of fourth-graders would know (Dale and Chall, 1948), contained in the Dale-Chall word-list. [6] It is calculated as follows:

$$\text{Raw Score} = 0.1579 \times \text{PDW} + 0.0496 \times \text{ASL}$$
$$\text{If PDW} > 5\% : \text{Adjusted Score} = \text{Raw Score} + 3.6365$$

All readability scores are represented as a US-grade level, where a higher grade means a more difficult text, except for the *Flesch Reading Ease*. The *Flesch Reading Ease* indicates a score between 0 (low readability) and 100 (high readability): a higher number means a more readable text. For this reason in most of our experiments the *Flesch Reading Ease* looks reversed with respect to the other measures (and is negatively correlated with them).

3 Results

Pearson's and Spearman's correlations between these five readability metrics and commonly used stylometric features show - as a sanity check - that readability measures capture aspects of novels' overall style. All measures are similarly correlated to sentence-length (naturally, being a base for all measures) but also to lexical diversity and compressibility, which measure, respectively, complexity at the word- and sequence-level. Moreover, the correlations with our "quality scores" show that readability is linked with the ones closer to popularity than to appreciation.

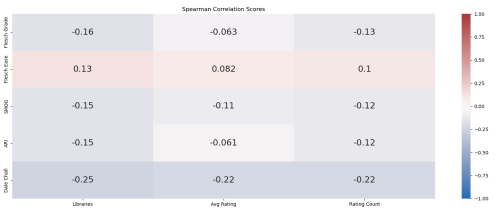


Figure 4: Correlations between quality standards and flavours of readability. All correlations are statistically significant.

Pearson's r, specifically in its significance testing, relies on the assumption of normally distributed data and it assumes that the two variables have a linear relationship, while Spearman's r correlation coefficient is non-parametric, meaning that, while it still assumes a monotonic relation

[5] All readability scores were extracted using the textstat package: https://pypi.org/project/textstat/

[6] See: https://countwordsworth.com/download /DaleChallEasyWordList.txt

between the two variables, it does not make strong assumptions on the shape of the data. For this reason, Spearman is probably the best overall measure for this study, as we have no reason to assume that all our measures are normally distributed (and some are evidently not, as can be seen in Figure 2). For these reasons, we will mainly credit the correlations observed through Spearman's r, although we report both in 2.

3.1 Readability and stylometrics

As readability measures are supposed to be measures of style, we compute their correlation with three core stylistic features - sentence length, lexical diversity[7] and textual compressibility[8] - that have been found linked to perceived literary quality in previous studies (van Cranenburgh and Bod, 2017; Crosbie et al., 2013; Maharjan et al., 2017; Wang et al., 2019). As can be seen in Figure 1, all readability measures have evident correlations with these three metrics, even though they don't necessarily compute them directly - for example, no readability measure computes text compressibility. However, while compressibility is not obviously correlated to readability, compressibility is a measure of redundancy or formulaicity: it appears that easier texts also have a tendency to be more sequentially repetitive. One readability measure, the new Dale-Chall, correlates with the simple length (word count) of the novels. This is a surprising effect, since, like the other measures, the new Dale-Chall is not length-dependent. As it is the only measure looking at the texts' lexicon through an index of difficult words, it seems to be picking on a tendency for longer books to have a slightly more complex vocabulary.

3.2 Relation with quality - GoodReads and libraries

As discussed before, we correlate readability with three possible proxies of perceived quality of novels: GoodReads' average ratings, GoodReads' rating count, and the number of libraries holding a given title according to WorldCat[9]. We could consider GoodReads' rating count to be a measure closer to the concept of popularity or fame, while GoodReads' average rating tells us about the appreciation of the title independently from how many readers it had. As can be seen in Figure 4, all of our readability measures show a degree of correlation with the number of library holdings and the GoodReads' rating count: more readable

Figure 5: The likelihood of being acquired by less than 100 libraries increases quite steadily with difficulty of reading (Spearman's rho 0.84), as the probability of appearing in more than 500 declines. Readability is here measured as Flesch-Kincaid Grade Level.

Figure 6: The probability of being rated by less than 100 users in Goodreads strongly correlates with the difficulty of the texts as measured, in this case, by the Flesch-Kincaid Grade Level.

[7]We operationalized lexical diversity as the type-token ratio (TTR) of a text, using a common method insensitive to text-length: the Mean Segmental Type-Token Ratio (MSTTR). MSTTR-100 represents the average TTR of local averages in 100-word segments of each text.

[8]Following van Cranenburgh and Bod (2017), for text compressibility, we calculated the compression ratio (original bit-size/compressed bit-size) using bzip2, a standard file-compressor.

[9]Naturally this selection remains arbitrary. Expanding to other measures of perceived quality is an ongoing process.

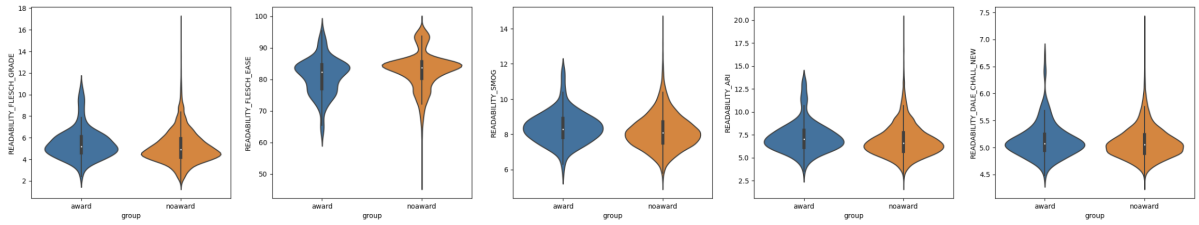


Figure 7: Flavours of readability and awards: overall distributions.

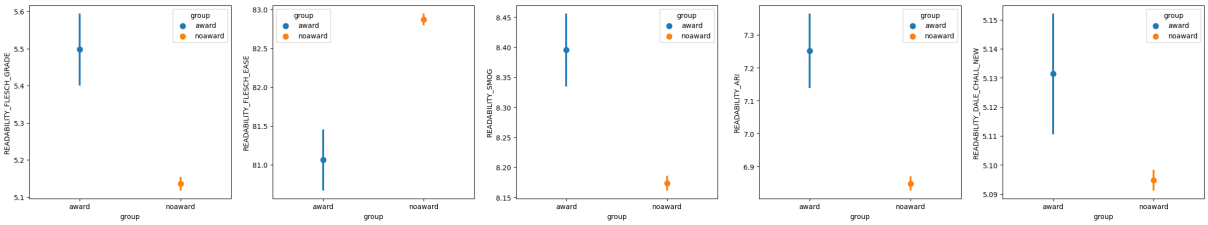


Figure 8: Flavours of readability and awards: mean value and standard error.

books tend to have more ratings and tend to be held by more libraries.

The average rating of titles on GoodReads, on the other hand, shows a significant correlation with only one of the measures, the *Dale-Chall* readability score, while it appears to have no link with the other four. Interestingly, the Dale-Chall score is the only measure that uses a precompiled list of words to estimate the number of difficult words in a text, instead of relying entirely on the features of the text at hand. While this could make it a more fragile measure (due to linguistic change and differences between genres) it appears to actually give it an increased modelling power for the tastes of GoodReads' average readers. It is worth mentioning that GoodReads' average ratings do not correlate, in our corpus, with the books' publication date - so a direct effect of language evolution on the measure's index can be excluded. Simplifying a bit, this points to the idea that the ease of vocabulary might relate to the average appreciation of a book as well as its fame, so that texts with a simpler lexicon, together with shorter sentences or words, are both more read and better liked.

In Figure 3 we show the relation of each readability measure with library holdings, average Goodreads ratings and number of Goodreads' ratings. As can be seen, we should interpret the results with some caution, as the relation might not be linear: it could be that the best interpretation of the relation between, for example, readability and library holdings is modelled with a curve rather than a straight line. Yet, it appears quite evident at a glance that the probability of being held by a large number of libraries, and of being rated by a large number of Goodreads users, decreases dramatically when the difficulty of the text increases beyond a certain level. As we show in Figure 5, the probability of being acquired by less than 100 libraries grows quite clearly with the text's difficulty, and the probability of being acquired by more than 500 decreases accordingly, with an interesting peak at a medium-low point of difficulty. The effect is even more evident when considering the probability of having less than 100 ratings on GoodReads, as appears in Figure 6. Appearing in 90 libraries is still a quite impressive measure of success, but the majority of the titles in the Chicago corpus goes beyond that threshold, as well as beyond the threshold of 100 user ratings on GoodReads, so the difference in probabilities seems to point to a relative decline in popularity or fame with the increase of the texts' surface complexity.

3.3 Relation with quality - literary awards

The second type of quality check we selected is a categorical one: whether or not a title was long-listed for one of three prestigious awards - the Pulitzer Prize, the National Book Award and the Hugo Award.

As we show in Figures 7 and 8, as well as in Table 3, the difference between long-listed books and non long-listed books in terms of readability is small but significant for almost all measures, with long-listed books are systematically harder to read

	Libs.	Rat. n.
Flesch grade	-0.16 (-0.1)	-0.06 (-0.06)
Flesch ease	0.13 (0.07)	0.08 (0.09)
SMOG	-0.15 (-0.1)	-0.11 (-0.11)
ARI	-0.15 (-0.01)	0.06 (-0.06)
New Dale-Chall	-0.25 (-0.2)	-0.22 (-0.2)
Flesch grade	0.84	0.83
Flesch ease	-0.4	-0.48
SMOG	0.76	0.81
ARI	0.73	0.71
New Dale-Chall	0.78	0.82

Table 2: On the upper part of the table, Spearman's r (Pearson's in parenthesis) for each readability flavour and quality measure. On the lower, Spearman's r with the probability of being in less than 100 libraries or having less than 100 ratings.

than their non-listed counterparts - again with the exception of the new Dale-Chall measure. Using this kind of quality proxy, we do not observe a value of reading ease but possibly its "dark side", such as perceived simplification or a reduced expressive power of novels.

It may not surprise that these different standards should exhibit different preferences and perspectives on quality. Literary awards are notoriously elitist, even, perhaps, in a way that is wanted by their readership: the committee of the Booker Prize was accused of populism in 2011 when announcing "readability" as a new criterion for the award (Clark, 2011).

	T-test	p-value
Flesch grade	3.78	0.0001
Flesch ease	-4.66	0.000005
SMOG	3.69	0.0002
ARI	3.6	0.0003
New Dale-Chall	1.8	0.07

Table 3: T-test and p-value for the difference between long-listed and non-listed titles for each readability measure. The only measure that does not fall under the formal threshold of statistical significance is the new Dale-Chall.

4 Conclusions and Future Works

Readability measures proved significantly consistent, both between each other and with other relevant stylometric features, when applied on modern and contemporary fiction. Their relation with different proxies of literary quality is intriguing: more popular works, in terms of number of ratings on GoodReads and in terms of libraries willing to hold a copy of the book, appear to have a correlation with readability, while the appreciation of readers alone (independently from their number) seems to hold almost no link with it, and long-listed titles have an inverse relation with readability, tending to prefer slightly more difficult prose on the readability metrics' scale. It can be argued that we are seeing the divide between high-brow and "popular" literature, but the lack of correlation with GoodReads average rating might point to a slightly more nuanced conclusion. It is worth noting that the only measure showing a meaningful correlation with all of the crowd-based quality metrics was the new Dale-Chall measure of readability, also the only one explicitly focusing on the presence of widely understood lexicon in a text, but it was also the only one showing no significant difference between long-listed and non long-listed titles. The only other measure having a correlation higher than 0.1 with average GoodReads' ratings was SMOG, which, while not using a list of hard words, considers "difficult words" in its own way in its computation, using the number of polysyllable words as a central element. If we were to draw rough conclusions from these observations, it would seem that surface-level simplicity of style in terms of words per sentence, characters per words, and similar metrics "helps" a text's popularity, but has nothing to do with its likelihood of being highly liked by its readers - and it even slightly hinders its possibilities of receiving a prestigious awards. In other words, surface-level simplicity improves a text's quality only if we equate it with popularity or fame. Similarly, looking at threshold-based probability distributions showed that indeed increasing the difficulty of the novels' style might hinder its diffusion across libraries and Goodreads' users. Using a more common vocabulary might also increase readers' appreciation of the text, but only when it comes to crowd-based measures. On the other hand, the correlations of average number of ratings and library holdings with readability measures do not appear linear or monotonic, meaning that there might also be a "point of balance" between too easy and too difficult, that maximizes the correlation with a novel's fame. The same might be true for the likelihood

of a novel being long-listed for one of the three awards we took into consideration.

Overall, readability seems to have an impact on different perceptions of literary quality, although its role and interaction with other features of the text remains to be defined. Another overarching point to observe from these findings is that there is a difference between crowd-based (GoodReads) and expert-based (awards) standards of literary quality in readability-level preference, which indicates that the criteria change across different quality-judgements, which suggetss that "literary quality" cannot be quantified reliably if it is reduced to a single golden standard. Further research points towards extending the set of correlations to more proxies of quality as well as more sophisticated stylometric measures to see whether interactions can provide a clearer picture of what we perceive as literary quality. Other further work could be to check the correlations of our measures with publication date: readability might depend on time, either in the sense of the evolution of the average novelistic style, overall language change, or even cultural selection, which would make the passage of time a particular form of "quality test" of its own accord.

References

Hélder Antunes and Carla Teixeira Lopes. 2019. Analyzing the Adequacy of Readability Indicators to a Non-English Language. In Fabio Crestani, Martin Braschler, Jacques Savoy, Andreas Rauber, Henning Müller, David E. Losada, Gundula Heinatz Bürki, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, volume 11696, pages 149–155. Springer International Publishing, Cham.

Judith Brottrager, Annina Stahl, Arda Arslan, Ulrik Brandes, and Thomas Weitin. 2022. Modeling and predicting literary reception. *Journal of Computational Literary Studies*, 1(1):1–27.

Jeanne S. Chall. 1947. This business of readability. *Educational Research Bulletin*, 26(1):1–13.

Jeanne S. Chall and Edgar Dale. 1995. *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books.

Alex Clark. 2011. Man Booker prize: This year's judges are betraying authors and their readers. *The Observer*.

Andreas van Cranenburgh and Rens Bod. 2017. A data-oriented model of literary language. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1228–1238, Valencia, Spain. Association for Computational Linguistics.

Tess Crosbie, Tim French, and Marc Conrad. 2013. Towards a model for replicating aesthetic literary appreciation. In *Proceedings of the Fifth Workshop on Semantic Web Information Management*, SWIM '13, New York, NY, USA. Association for Computing Machinery.

Edgar Dale and Jeanne S. Chall. 1948. A formula for predicting readability. *Educational Research Bulletin*, 27(1):11–28.

Edgar Dale and Jeanne S. Chall. 1949. The concept of readability. *Elementary English*, 26(1):19–26.

William Dubay. 2004. *The Principles of Readability*. Impact Information.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32:221–233.

Craig L. Garthwaite. 2014. Demand spillovers, combative advertising, and celebrity endorsements. *American Economic Journal: Applied Economics*, 6(2):76–104.

Corina Koolen, Karina van Dalen-Oskam, Andreas van Cranenburgh, and Erica Nagelhout. 2020. Literary quality in the eye of the Dutch reader: The national reader survey. *Poetics*, 79:1–13.

Suraj Maharjan, John Arevalo, Manuel Montes, Fabio A. González, and Thamar Solorio. 2017. A multi-task approach to predict likability of books. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1217–1227, Valencia, Spain. Association for Computational Linguistics.

Suraj Maharjan, Sudipta Kar, Manuel Montes, Fabio A. González, and Thamar Solorio. 2018. Letting emotions flow: Success prediction by modeling the flow of emotions in books. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Volume 2, Short Papers*, pages 259–265, New Orleans, Louisiana. Association for Computational Linguistics.

Claude Martin. 1996. Production, content, and uses of bestselling books in quebec. *Canadian Journal of Communication*, 21(4).

Harry G. McLaughlin. 1969. Smog grading: A new readability formula. *Journal of Reading*, 12(1):639–646.

Mahdi Mohseni, Christoph Redies, and Volker Gast. 2022. Approximate entropy in canonical and non-canonical fiction. *Entropy*, 24(2):278.

Lisa Nakamura. 2013. "Words with friends": Socially networked reading on Goodreads. *PMLA*, 128(1):238–243.

Janice Redish. 2000. Readability formulas have even more limitations than Klare discusses. *ACM J. Comput. Doc.*, 24(3):132–137.

Lucius A. Sherman. 1893. *Analytics of Literature: A Manual for the Objective Study of English Prose and Poetry*. Athenaeum Press. Ginn.

Sanja Stajner, Richard Evans, Constantin Orasan, and Ruslan Mitkov. 2012. What can readability measures really tell us about text complexity? In *Proceedings of Workshop on natural language processing for improving textual accessibility*, pages 14–22, Istanbul, Turkey. Association for Computational Linguistics.

Xindi Wang, Burcu Yucesoy, Onur Varol, Tina Eliassi-Rad, and Albert-László Barabási. 2019. Success in books: Predicting book sales before publication. *EPJ Data Science*, 8(1):31.

Detection and attribution of quotes in Finnish news media: BERT vs. rule-based approach

Maciej Janicki and **Antti Kanner** and **Eetu Mäkelä**
Department of Digital Humanities
University of Helsinki
Unioninkatu 40, 00170 Helsinki, Finland
firstname.lastname@helsinki.fi

Abstract

We approach the problem of recognition and attribution of quotes in Finnish news media. Solving this task would create possibilities for large-scale analysis of media wrt. the presence and styles of presentation of different voices and opinions. We describe the annotation of a corpus of media texts, numbering around 1500 articles, with quote attribution and coreference information. Further, we compare two methods for automatic quote recognition: a rule-based one operating on dependency trees and a machine learning one built on top of the BERT language model. We conclude that BERT provides more promising results even with little training data, achieving 95% F-score on direct quote recognition and 84% for indirect quotes. Finally, we discuss open problems and further associated tasks, especially the necessity of resolving speaker mentions to entity references.

1 Introduction

The recognition of quotes and reported speech is an important step towards the computational analysis of news media articles. It allows us to measure, on a large scale, who is given voice and how much, how opposing or competing views are presented alongside each other, as well as how the language of the quoted sources differs from the language of the journalistic reporting. In case of the Finnish news media, such analyses have recently been attempted by (Koivunen et al., 2021; Seuri et al., 2021). On the other hand, Suomen Kuvalehti et al. (2021) have studied politicians' visibility in the media based on the mentions of their names.

In the present paper, we focus on the technical task of recognizing direct and indirect quotes in the Finnish news media texts. The task can be illustrated with the following example:

> **Sipilän** mukaan *lakiehdotuksia ollaan tuomassa eduskuntaan helmikuussa.*
>
> According to **Sipilä**, *bill proposals will be brought to the parliament in February.*

Such relations consists of three elements: the **cue** 'mukaan' ('according to') indicates an indirect quote, in which the **source** (Juha Sipilä, the Finnish prime minister 2015–2019) says the text referred to as **proposition**, or **quotation span**.[1] A complete approach for quote detection and attribution would solve the following tasks:

1. Detecting quotation spans.
2. Attributing quotation spans to the source mention in the text (which might also span multiple tokens).
3. Linking source mentions to entity identifiers (including coreference resolution and lemmatization).

We will present methods for solving tasks 1 and 2, while discussing 3 as subject for further work.

Most existing work for this task deals with English, while occasionally other Germanic or Romance languages have been considered. Compared to that, Finnish presents challenges due to a rich morphology and free word order. Those can largely be dealt with by the advanced NLP tools that we are using (either a dependency parser pipeline or BERT), but they rule out the usage of simpler pattern-based methods and remain a possible source of errors even for state-of-the-art NLP.

[1] We follow Pareti (2015)'s convention of marking the quotation span in cursive, the source in bold, and underlining the cue.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 52–59

We describe the process of collecting and annotating a gold standard corpus in Sec. 3. Further, in Sec. 4, we describe two different automatic approaches: a rule-based one, amounting to matching certain grammatical structures in dependency-parsed text, as well as a machine learning one, which utilizes the state-of-the-art neural language model BERT. The corpus and the code for both methods are publicly available.[2] [3] [4]

Our initial intuition was that dependency parsing provides enough information to recognize quotes with simple pattern matching. Another reason to implement this approach was that it did not need training data, which was at first unavailable for us. However, the final comparison revealed that the BERT-based model outperformed the rule-based even with little training data. The results of this experiment are described in Sec. 5.

2 Related Work

To our knowledge, the most similar work to ours has been done by Silvia Pareti and colleagues (Pareti et al., 2013; Pareti, 2015, 2016), who annotated a corpus of attribution relations for English and experimented with machine learning models for recognizing such relations. For the latter they applied classification algorithms – CRF, k-NN, logistic regression – working on data enriched with linguistic features, which was state-of-the art in NLP at the time. However, Scheible et al. (2016) have criticized the choice of CRFs for quote detection because of the Markov assumption they make. More recently, Papay and Padó (2019) presented a neural LSTM-based model for recognizing quotations, but without attribution. Brunner et al. (2020) compare different embedding-based models (including BERT) on the task of recognizing types of speech, which includes direct and indirect quotes.

As to Nordic languages, a rule-based approach for Norwegian has been presented by Salway et al. (2017). It utilizes a dependency parser and a list of speech verbs. From among other languages, Quintão (2014) used a machine learning method on Portuguese news corpora, while Pouliquen et al. (2007) used a rule-based approach for multiple European languages.

Muzny et al. (2017) present a method for quote attribution. They thus start with quotation spans already recognized and perform two tasks: 1) attributing a quote to a speaker mention in the text, 2) linking the speaker mentions into entities. They use a rule-based strategy on top of tools performing dependency parsing and coreference resolution. They have also released a corpus of quote attributions consisting of three novels in English.

Although not dealing exactly with quote detection, Padó et al. (2019) provide a prominent example of computational analysis of political discourse using modern NLP methods. They use various neural models (including BERT) to detect claims and attribute them to actors, with the goal of modeling the discourse as a network of relations between actors and claims. Automatic quote detection could be a useful element of such a larger system as well.

3 Dataset and Annotation

The annotation process consisted of two parallel tasks: marking quotations and linking together chains of co-referencing expressions denoting people, institutions and other human-like actors present in the documents. Both annotation tasks were conducted using the WebAnno platform (Eckart de Castilho et al., 2016), by which each annotator was assigned their documents and by which the annotation itself was done. The annotation guidelines were written beforehand and further developed after a test run.

The quotation detection annotation consisted of 1) marking the span in the text containing the content of the quote, 2) marking the speech act verb (if present), 3) marking the source of the quotation (if present), and 4) noting whether the quote was direct or indirect. The task was relatively straightforward, as all annotators were students with at least a minor degree in linguistics.

The project employed 10 annotators. Four of them were recruited in an earlier phase and annotated a test data set of 40 articles. After the test run, the guidelines were improved based on both inter-annotator agreement scores and feedback from the annotators, in accordance with the standard linguistic annotation methodology (Artstein, 2017). The inter-annotator agreement scores (Fleiss' κ) were between 0.77-0.8, which we deemed sufficient to consider the annotations consistent. The workload was balanced so that the 6

[2] https://github.com/hsci-r/fi-quote-coref-corpus

[3] https://github.com/hsci-r/flopo-quote-detection

[4] https://github.com/hsci-r/flopo-quotes-bert

other annotators who were recruited at the later stage annotated more articles to compensate for the test run. The annotators worked independently on the WebAnno platform.

The articles were sampled from a database containing the metadata for the online media sources and the sampled lists of articles were then scraped using a web crawler (Mäkelä and Toivanen, 2021) and automatically pre-processed to CONLL format containing lemmatization, part-of-speech and dependency taggings using Turku Neural Parser (Kanerva et al., 2018). We used four sources for the articles: YLE (the Finnish national broadcasting company), Helsingin Sanomat (the most popular daily newspaper), Iltalehti (an evening tabloid) and STT (the Finnish news agency), covering different kinds of media texts wrt. length and style. The total number of articles annotated was 1500. Except for the common part mentioned above, the remaining 1460 articles were assigned to one annotator each at the second stage.

4 Methods

4.1 Rule-based approach

The input to the rule-based quote detection engine is text with linguistic annotations obtained from the Turku Neural Parser (Kanerva et al., 2018).[5] The parser performs the following tasks: tokenization, lemmatization, part-of-speech and morphological tagging, and dependency parsing.

The first stage of quote recognition is recognizing syntactic structures that typically introduce a quote (Table 1). Rules 1-2 describe the very common structures like 'X says that Y' and 'Y, says X', respectively. Rules 3-4 describe structures of the type: 'according to X, Y' and 'in X's opinion, Y'. In such structures, the source and cue can be positioned differently relatively to the proposition: before, after, or even inside it (see the example for rule 4). In the latter case, we allow annotating the cue and source as part of the proposition to avoid discontinuous propositions. Finally, rule 5 is characteristic for Finnish: it captures the construction '*says* + active participle', e.g. *sanoo olevansa* 'says that he is', or *sanoo tehneensä* 'says that he did'. This construction does not use the word *että* 'that'.

In the rules where the cue is a verb (1, 2 and 5), the verb *sanoa* 'to say' can be substituted by any other speech act verb, e.g. *kertoa* 'to tell', *korostaa* 'to emphasize', *kuitata* 'to sum up' etc. We initially prepared a list of speech act verbs manually, then used a word2vec model to expand it with automatically generated synonyms, which were again filtered manually. The final list consisted of 73 verbs.

Once the source-cue-proposition triplets are recognized, the proposition texts can typically be extracted by taking the dependency subtree under the token marked as proposition. However, further post-processing is needed for quotes consisting of multiple sentences. For example in Table 1, the example for rule 2 is clearly the last sentence of a multi-sentence quote. In order to expand the matches to multi-sentence quotes, we use two rules:

1. If the paragraph containing the match starts with a hyphen – extend the quote to the beginning of the paragraph. This is because long direct quotes are typically formatted as separate paragraphs.

2. If there is a quotation mark between the cue and the proposition head – extend the quote backwards to the matching quotation mark.

In both these cases, the quote is classified as direct, as it is marked with quotation markers. Matches that do not fulfill the above conditions are classified as indirect.

Finally, we use an additional rule to detect 'free-standing' direct quotes encompassing entire paragraphs. These do not necessarily contain a source attribution (like ', says X') because the source might be already clear from the context. Thus, we detect remaining paragraphs that either start with a hyphen or are enclosed in quotation marks, as direct quotes. For the attribution we currently use a naïve strategy of attributing them to the same source as the previous quote in the text (if present). This works in a lot of cases because the quotes usually follow a structure in which a whole-paragraph direct quote is introduced by a preceding sentence containing an indirect quote, like in the following example:

[5] A reviewer has plausibly remarked that using the dependency parser available in spaCy could simplify the architecture. We have not evaluated the impact of this change on performance, as at the time of implementing the method Turku Neural Parser was considered state-of-the-art for Finnish and, unlike spaCy, the Turku parser was applied in various other ways in the project context. However, the rules are coded in the spaCy DependencyMatcher format, so they can easily be tried on spaCy output as well.

According to Lindberg, approximately every third pet is overweight.

– We do have a lot of work on that.

The rules from Table 1 are implemented using the spaCy library class `DependencyMatcher`[6] which offers a declarative language to express the rules and good performance. The post-processing code is implemented in Python.

4.2 BERT model

The machine learning model is realized as two token classification heads on top of BERT – a neural language model based on the transformer architecture (Devlin et al., 2019). We use the model pretrained on Finnish data by Virtanen et al. (2019).

The first classification head recognizes and classifies spans of quoted text (propositions). The labeling follows the IOB schema and the class label encodes whether the quote is direct or indirect, as well as the relative position of the speaker mention to the quoted text. The latter is expressed as one of the symbols: +, − or = and a number 1-4. The symbol describes whether the speaker is mentioned after (+), before (−) or inside (=) the proposition, while the number signifies, which recognized entity is the speaker. For example, the class label `B-DIRECT+2` denotes the *beginning* (`B-`) of a *direct* quote, the source of which is the *second* recognized entity *after* the quote. A special label `O0` signifies that the source of the quote is not marked.

The second classification head recognizes the entities, i.e. elements of coreference chains. It has just one class encoded in the IOB schema and does not perform the linking of entities into chains.

An example of sequence annotation is shown in Table 2. It shows the following sentence:

> *Kansainvälinen rikostuomioistuin aikoo määrätä Sudanin presidentin Omar al-Bashirin pidatettäväksi*, kertoo **sanomalehti New York Times.**
>
> *The International Criminal Court is intending to issue an arrest warrant on Sudan's president Omar al-Bashar*, **the newspaper New York Times** reports.

There are three entities in the sentence: 'The International Criminal Court', 'Sudan's president Omar al-Bashar' and 'the newspaper New York Times' – their annotations on the token level are encoded on the 'entity' layer. The 'quote' layer encodes an indirect quote, which is attributed to the first entity following the quote (hence, `+1`).

5 Evaluation

For the evaluation experiments we use a roughly 80-20 split of the data by taking the data provided by 2 annotators as evaluation set and the remaining 8 annotators as training set. The dataset sizes are summarized in Table 3. We compare both methods on the task of quote recognition (with and without direct/indirect classification) and attribution.

Quote detection. The results of quote span detection without taking into account the direct-indirect distinction are shown in Table 4. On the other hand, the direct-indirect breakdown is shown in Table 5, where misclassifications (identifying a direct quote as an indirect one or vice versa) were counted as both a false positive and a false negative. We exclude punctuation tokens from the evaluation as especially the commas and periods on the boundaries of quotes might have been inconsistently annotated, and their inclusion in the quote is irrelevant.

Both settings show a clear advantage of the BERT model. In case of direct quotes, the rules for recognizing them are quite rigid. Furthermore, they can suffer from paragraph segmentation errors and misplaced or incidental quotation marks (e.g. 'scare quotes'). This explains the lower recall of the rule-based method.

Indirect quotes have proven more challenging to the rule-based method as well. This can be to a variety of reasons: missing speech act verbs, incorrectly identifying quote spans based on syntactic criteria (also affected by parser, tagger and sentence segmentation errors), or uncommon structures not covered by the rules. Moreover, rule 3 ('according to') has a tendency to produce false positives, e.g. something being described 'according to the plan'.

In general, the BERT model has shown to be more flexible wrt. the often unpredictable nature of text data, and does not suffer from the error propagation through the NLP pipeline.

Attribution. The evaluation of attribution is problematic because of the fact that our dataset was not annotated with the BERT model in mind.

[6] `https://spacy.io/api/dependencymatcher`

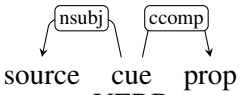
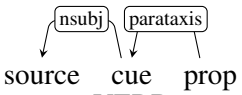
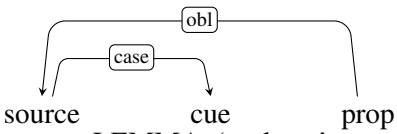
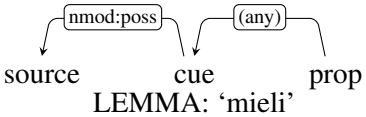
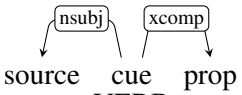
No.	schema	example
1		**Malinen** sanoo, että *hän ei tule esittämään liiton hallitukselle yhdenkään sopimuksen hyväksymistä.* **Malinen** says that *he will not propose accepting even a single motion of agreement to the union's board.*
2		*Siksi mekin lähdimme näihin neuvotteluihin mukaan*, **Mäkynen** sanoo. *This is why we also joined these negotiations*, **Mäkynen** says.
3		**Sipilän** mukaan *lakiehdotuksia ollaan tuomassa eduskuntaan helmikuussa.* According to **Sipilä**, *bill proposals will be brought to the parliament in February.*
4		*Suomen vaikeista ongelmista talous on* **presidentin** mielestä *helpompi.* *From Finland's most difficult problems, the economy is* in **the president's** opinion *easy.*
5		**Orpo** sanoo *olevansa valmis poikkeuksellisiin keinoihin ja jopa lainmuutoksiin [. . .].* **Orpo** says *that he is ready for exceptional measures and even legistative changes [. . .].*

Table 1: The manually constructed rules for detecting quote-like syntactic structures.

word	quote	entity
Kansainvälinen	B-INDIRECT+1	B
rikostuomioistuin	I-INDIRECT+1	I
aikoo	I-INDIRECT+1	O
määrätä	I-INDIRECT+1	O
Sudanin	I-INDIRECT+1	B
presidentin	I-INDIRECT+1	I
Omar	I-INDIRECT+1	I
al-Bashirin	I-INDIRECT+1	I
pidätettäväksi	I-INDIRECT+1	O
,	O	O
kertoo	O	O
sanomalehti	O	B
New	O	I
York	O	I
Times	O	I
.	O	O

Table 2: An example of sequence annotation for the BERT model.

	training	evaluation
articles	1,172	287
sentences	22,949	5,097
tokens	252,006	59,076
quotes	3,854	984

Table 3: The sizes of datasets used in experiments.

method	Pr	Re	F1
rule-based	.85	.78	.82
BERT	.92	.90	.91

Table 4: Results of quotation span detection without classification.

	indirect			direct		
method	Pr	Re	F1	Pr	Re	F1
rule-based	.75	.66	.70	.93	.86	.89
BERT	.84	.84	.84	.96	.94	.95

Table 5: Results of quotation span detection and direct/indirect classification.

Thus, we present it as our best attempt given the current possibilities, but recognize the need for further work in this regard.

The annotated data assigns each quote to a single token representing the mention of the quote's source in the text. If the source is represented by a longer phrase, the syntactic head (wrt. dependency parsing) of this phrase should be selected according to the annotation guidelines. On the other hand, mentions of quote sources are typically entities annotated as parts of coreference chains, and thus the entire span is marked for the purpose of coreference annotation. Thus, by combining the quote and coreference annotations, we are able to obtain a span-to-span attribution relation for most cases. The exception are cases in which the quoted entity is mentioned only once in the article, and thus not annotated as a coreference chain.

Although the BERT model outputs sources as entity spans, the rule-based model points to a single token – the syntactic head, similarly to the gold standard annotation. In order to make the results comparable, we reduced the output of the BERT model to the first token of the span, and then evaluated a source annotation as correct if it either points to exactly the same token as the gold standard, or if it points to a token within the same coreference span. Thus, the model's ability to correctly identify the entire span is currently not evaluated, as it is not implemented in the rule-based method.

Table 6 presents results of the attribution evaluation in terms of the number of gold-standard quote tokens with **cor**rectly and **inc**orrectly recognized source, as well as **unrec**ognized source. The latter case occurs if either the token is not recognized as a quote at all, or it is recognized but without identifying the source. We report the accuracy as the ratio of correctly identified to all tokens.

The results indicate a small advantage of the rule-based model. In both cases, the main source of errors are the unrecognized annotations, rather than the incorrect ones. For the rule-based model this is typically due to quotes not being recognized at all (see low recall in Table 4), while for the BERT model there is a large amount of correctly identified quotes, for which the source could not be found. Of the 1990 recognized quotes, 646 (32%) are reported without source, compared to 13% (218/1633) for the rule-based model. The BERT model's ability to identify the source depends on the entity detection, for which the training data is incomplete (derived from coreference annotations only). Further, the model processes the text paragraph by paragraph and thus does not find a source mention that is outside of the paragraph containing the quote. These problems offer room for improvement in further work, and thus it can be expected that the BERT model will eventually outperform the rule-based one also in attribution.

method	cor	inc	unrec	accuracy
rule-based	7889	774	4996	.58
BERT	7554	767	5338	.55

Table 6: Results of attribution.

6 Discussion and Further Work

Although we regard the work presented in the previous sections as a complete solution to a well-delimited problem, we see some potential for both incremental improvements, as well as work on further related tasks, that will be addressed in the future.

Entity annotation and detection. While designing our annotation project, we did not anticipate that a machine learning quote detection model will need to also detect entities that the quotes can be attributed to. We intended the coreference annotation to be used only in the further step (entity resolution). In result, entities that are mentioned only once were not annotated. The corpus could be improved by ensuring that at least tokens assigned as source to a quote are also annotated as an entity. This is expected to improve the BERT model's performance on entity detection, and thus quote attribution.

Entity resolution. While some works treat the problem of quote attribution to speaker mention in the text and entity resolution jointly (e.g. Muzny et al., 2017), in our opinion entity resolution is a complex task that is best treated separately. In addition to coreference resolution within one document, also matching the entities across documents could be considered there.

Coreference resolution can be done with BERT with state-of-the-art accuracy (Joshi et al., 2019). However, the setup is complicated as coreferences are typically long-range relations, so a sliding win-

dow approach needs to be used to mitigate BERT's limitation in text size. Furthermore, modeling relations with a neural model is not straightforward.

A related problem is that nested entities are possible and might be relevant, e.g.:

> [[[Viron] metallityöväen liiton] puheenjohtaja Endel Soon]
>
> [[[Estonia]'s metal workers' union]'s chairman Endel Soon]

In such case, coreferences and other quotes might also refer to the inner entities 'Estonia' or 'Estonia's metal workers' union'. For the present work, we disregarded nested entities as locally the outermost entity is typically the source of the quote it stands next to.

7 Conclusion

We have presented two methods for recognition of quotes in Finnish news media, along with an annotated corpus for training and evaluation. To our knowledge, our solution is the first one proposed for Finnish. We hope that the progress achieved on this task will facilitate more detailed large-scale quantitative analysis of voices in the Finnish news media.

References

Ron Artstein. 2017. *Handbook of Linguistic Annotation*, chapter Inter-annotator agreement.

Ann Brunner, Ngoc Duyen Tanja Tu, Lukas Weimer, and Fotis Jannidis. 2020. To bert or not to bert - comparing contextual embeddings in a deep learning architecture for the automatic recognition of four types of speech, thought and writing representation. In *SwissText/KONVENS*.

Richard Eckart de Castilho, Éva Mújdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. 2016. A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84, Osaka, Japan.

Jacob Devlin, Ming-Wei Chang, Kanton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Mandar Joshi, Omer Levy, Daniel S. Weld, and Luke Zettlemoyer. 2019. Bert for coreference resolution: Baselines and analysis. In *EMNLP 2019*.

Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku neural parser pipeline: An end-to-end system for the conll 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Anu Koivunen, Antti Kanner, Maciej Janicki, Auli Harju, Julius Hokkanen, and Eetu Mäkelä. 2021. Emotive, evaluative, epistemic: a linguistic analysis of affectivity in news journalism. *Journalism*, 22(5):1190–1206.

Grace Muzny, Michael Fang, Angel X. Chang, and Dan Jurafsky. 2017. A two-stage sieve approach for quote attribution.

Eetu Mäkelä and Pihla Toivanen. 2021. Finnish media scrapers. *Journal of Open Source Software*, 6(68):3504.

Sebastian Padó, André Blessing, Nico Blokker, Erenay Dayanik, Sebastian Haunss, and Jonas Kuhn. 2019. Who sides with whom? towards computational construction of discourse networks for political debates. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2841–2847.

Sean Papay and Sebastian Padó. 2019. Quotation detection and classification with a corpus-agnostic model. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 888–894, Varna, Bulgaria. INCOMA Ltd.

Silvia Pareti. 2015. *Attribution: A Computational Approach*. Ph.D. thesis, University of Edinburgh.

Silvia Pareti. 2016. Parc 3.0: A corpus of attribution relations. In *LREC*.

Silvia Pareti, Tim O'Keefe, Ioannis Konstas, James R. Curran, and Irena Koprinska. 2013. Automatically detecting and attributing indirect quotations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 989–999.

Bruno Pouliquen, Ralf Steinberger, and Clive Best. 2007. Automatic detection of quotations in multilingual news. In *Proceedings of Recend Advances in Natural Language Processing*, pages 487–492, Borovets, Bulgaria.

Marta Quintão. 2014. Quotation attribution for portuguese news corpora.

Andrew Salway, Paul Meurer, Knut Hofland, and Øystein Reigem. 2017. Quote extraction and attribution from norwegian newspapers. In *Proceedings of the 21st Nordic Conference of Computational Linguistics*, pages 293–297, Gothenburg, Sweden.

Christian Scheible, Roman Klinger, and Sebastian Padó. 2016. Model architectures for quotation detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1736–1745.

Olli Seuri, Riikka Era, Anu Koivunen, Maciej Janicki, Pihla Toivanen, Julius Hokkanen, and Eetu Mäkelä. 2021. Uutisvuon hallitsija: Uutismedia kiky-kamppailussa 2015–2016. *Politiikka : Valtiotieteellisen yhdistyksen julkaisu*, 63(3):233–259.

Suomen Kuvalehti, Eetu Mäkelä, and Pihla Toivanen. 2021. Vuosi valokeilassa: Kuka sai medialta huomiota? Kuka jäi varjoon? Suomen kuvalehti selvitti tutkijoiden kanssa, miten kansanedustajat näkyivät neljässä suuressa uutismediassa vuonna 2020.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for Finnish.

Dyslexia Prediction from Natural Reading of Danish Texts

Marina Björnsdóttir
IT University of Copenhagen
University of Copenhagen
marina.bjorns@gmail.com

Nora Hollenstein
University of Copenhagen
University of Zurich
nora.hollenstein@hum.ku.dk

Maria Barrett
IT University of Copenhagen
mbarrett@itu.dk

Abstract

Dyslexia screening in adults is an open challenge since difficulties may not align with standardised tests designed for children. We collect eye-tracking data from natural reading of Danish texts from readers with dyslexia while closely following the experimental design of a corpus of readers without dyslexia. To our knowledge, this is the first attempt to classify dyslexia from eye movements during reading in Danish. We experiment with various machine-learning methods, and our best model yields a 0.85 macro F1 score.

1 Introduction

Dyslexia is a learning disorder of neurological origin that reportedly affects about 10-20% of the world population (Rello and Ballesteros, 2015; Kaisar, 2020). It involves difficulties with reading, spelling, and decoding words, and is not related to intelligence (Perera et al., 2018; Rauschenberger et al., 2017). Detecting dyslexia as early as possible is vital, as the disorder can lead to many negative consequences that can be mitigated with proper assistance. These include low self-esteem and high rates of depression and anxiety (Perera et al., 2018; Schulte-Körne, 2010). There are qualitative studies suggesting that living with an undiagnosed learning disorder leads to frustrations (Kong, 2012), feelings of being misunderstood (Denhart, 2008), and of failure, (Tanner, 2009). Being diagnosed with a learning disorder as an adult has been reported to lead to a sense of relief (Arceneaux, 2006), validation (Denhart, 2008; Kelm, 2016) and liberation (Tanner, 2009; Kong, 2012). Dyslexia can be difficult to diagnose due to its indications and impairments occurring in varying degrees (Eckert, 2004), and is therefore often recognised as a *hidden disability* (Rello and Ballesteros, 2015). Popular methods of detecting dyslexia usually include standardised lexical assessment tests that involve behavioural aspects, such as reading and spelling tasks (Perera et al., 2018). Singleton et al. (2009) explain that computerised screening methods have been well-established for children in the UK, but developing such tests for adult readers with dyslexia is exceptionally challenging as adults with dyslexia may not show obvious literacy difficulties that align with what standardised tests distinguish as dyslexic tendencies. For one thing, dyslexia is experienced differently from person to person. Still, also, most adults with dyslexia have developed strategies that help them disguise weaknesses and may thus remain unnoticed and result in false-negative tests (Singleton et al., 2009).

Less frequently used methods are eye tracking during reading or neuroimaging techniques such as (functional) magnetic resonance imaging, electroencephalogram, brain positron emission tomography, and magnetoencephalography methods (Kaisar, 2020; Perera et al., 2018). These models are yet under experimental development and are currently not used for screening dyslexia (Perera et al., 2018). A small body of studies investigates dyslexia detection using eye tracking with the help of machine-learning techniques outlined in §2.4. Compared to neuroimaging techniques, eye tracking is more affordable and faster to record and its link to online text processing is well established (Rayner, 1998). Using eye-tracking records for dyslexia detection does not necessarily require readers to respond or perform a test but merely objectively observes the reader during natural reading (Benfatto et al., 2016). Although eye-tracking experiments are often limited to a relatively small number of participants compared to computerized tools, the method typically produces many data points from each participant.

The purpose of the current paper is twofold:

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 60–70

1) We provide a dataset from participants with dyslexia reading Danish natural texts. This dataset uses the same experimental design as the CopCo corpus by Hollenstein et al. (2022), which allows us to compare the eye movement patterns from readers with dyslexia to those without from CopCo. 2) We train the first machine learning (ML) classifiers for dyslexia prediction from eye movements in Danish. The data is available as raw gaze recording, fixation-level information, and word-level eye tracking features.[1] The code for all our experiments is also available online.[2]

2 Related Work

2.1 Dyslexia Screening in Denmark

In 2015, The Ministry of Children and Education in Denmark launched a national electronic dyslexia test, Ordblindetesten 'the Dyslexia Test'. The test is a screening method for children, youths, and adults speculated to have dyslexia. It is accessible through educational institutions and is performed under the observation of a supervisor (Centre for Reading Research et al., 2020). It consists of three multiple-choice subtests, performed electronically, that focus on phonological decoding abilities. The result is 'not dyslexic,' 'uncertain phonological decoding,' or 'dyslexic.' The official instruction strictly denies the uncertain group to be dyslexic[3] and therefore not entitled to dyslexia support. But they may benefit from other support and are subject to further assessment, e.g., text comprehension, reading speed, spelling, and vocabulary tests appropriate for the examinee's age and educational requirements (Centre for Reading Research et al., 2020). To this end, Helleruptesten "The Hellerup Test" is used by educational institutions for adults.[4]

2.2 Danish as a Target Language

Similar studies on dyslexia detection with ML classification include experiments with Chinese (Haller et al., 2022), Swedish (Benfatto et al., 2016), Spanish (Rello and Ballesteros, 2015), Greek (Asvestopoulou et al., 2019), Arabic (Al-Edaily et al., 2013) and Finnish (Raatikainen et al., 2021) as their target languages. However, the diagnostic characteristics of dyslexia may differ depending on the transparency of the language. In early research, De Luca et al. (1999) reported that the regular spelling-sound correspondences in languages of transparent orthographies, e.g., German and Italian, dim phonological deficits. Phonological deficits of individuals with dyslexia are clearer in languages with irregular, non-transparent orthographies (Smyrnakis et al., 2017).

Danish is a language with a highly non-transparent orthography. It has been shown that overall adult reading comprehension skills are poorer in Danish than in other Nordic languages (Juul and Sigurdsson, 2005). The lack of spelling-sound correspondence in Danish indicates that the Danish language holds excellent value for investigating dyslexia detection based on two main reasons: Firstly, the combination of the non-transparent orthography of the Danish language and eye movement patterns could potentially reveal more apparent indications of dyslexia through the selected features that have proven to be relevant for dyslexia detection in other languages, which can be favourable in further research on, e.g., the development of assistive tools and technologies. Enabling a direct comparison between eye-tracking data from adults with dyslexia and adults without dyslexia with Danish as the target language will provide beneficial insights into reading dyslexic patterns, which can be favourable in further research, e.g., the development of assistive tools and technologies. Secondly, the fact that reading comprehension skills are proven to be poorer in Danish than in other Nordic languages highlights the necessity of proper assistance and recognition for individuals with dyslexia in Denmark.

2.3 Dyslexia and Eye Movements

Tracking eye movements during natural reading reveals information on fixations (relatively still gaze on a single location) and saccades (rapid movements between fixations). Studies (Rayner, 1998; Henderson, 2013) have substantiated that information on eye movements during reading contains characterizations of visual and cognitive processes that directly impact eye movements. These are also strongly related to identifying information about, e.g., attention during reading, which is highly correlated with saccades (Rayner, 1998).

[1] https://osf.io/ud8s5/

[2] https://github.com/norahollenstein/copco-processing

[3] https://www.spsu.dk/for-stoettegivere/elever-og-studerende-med-usikker-fonologisk-kodning

[4] from Vestegnen VUC, an educational institution that provides education for students with dyslexia

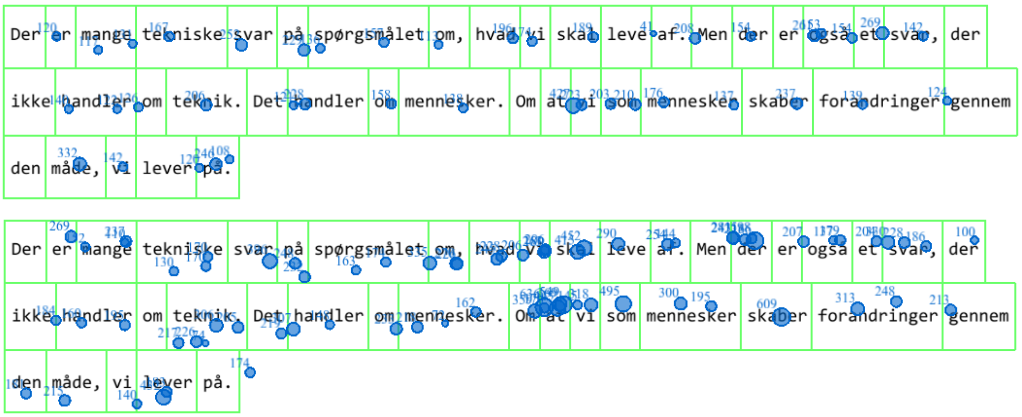


Figure 1: Fixations recorded from a reader without dyslexia (above) and a reader with dyslexia (below) when reading the same sentence. Numbers indicate duration in ms.

As Henderson (2013) phrases it, "eye movements serve as a window into the operation of the attentional system."

Previous studies have repeatedly shown that readers with dyslexia show more fixations and regressions, longer fixation durations, and shorter and more numerous saccades than readers without dyslexia (Pirozzolo and Rayner, 1979; Rayner, 1986; Biscaldi et al., 1998). This was already discovered by Rubino and Minden (1973) and later work discussed whether this was the cause or effect of dyslexia with evidence on both sides, e.g., Pirozzolo and Rayner (1979); Pavlidis (1981); Eden et al. (1994); Biscaldi et al. (1998). Most recent studies acknowledge that the movements reflect a dyslexic reader's difficulties with processing language. (Fischer and Weber, 1990; Hyönä and Olson, 1995; Henderson, 2013; Rello and Ballesteros, 2015; Benfatto et al., 2016; Raatikainen et al., 2021), and Rayner (1998) who echo an earlier study (Rayner, 1986) state that eye movements are not the cause of slow reading but rather reflect the more time-consuming cognitive processes. These insights from psycholinguistics motivate the feature selection for this work.

2.4 ML-based Dyslexia Detection from Gaze

Recent evidence shows that ML-based methods can be used for dyslexia detection in children, e.g., Christoforou et al. (2021); Nerušil et al. (2021). This section is, however, limited to ML-based methods for dyslexia detection in adults. Prior studies that facilitate the investigation of dyslexia detection with the help of machine learning classification on eye-tracking data have concluded that support vector machines (SVM's) is of great advantage (Rello and Ballesteros, 2015; Benfatto et al., 2016; Prabha and Bhargavi, 2020; Asvestopoulou et al., 2019; Raatikainen et al., 2021). Rello and Ballesteros (2015) used an SVM for dyslexia detection based on eye-tracking recordings from readers with and without dyslexia, which resulted in an accuracy of 80.18%. Benfatto et al. (2016); Prabha and Bhargavi (2020) achieved accuracy scores of 95.6% and 95% respectively on the same dataset using SVM variations.

With Greek as their target language, Smyrnakis et al. (2017) propose a method with two parameters for dyslexia detection: word-specific and non-word-specific. Non-word-specific features consisted of fixation duration, saccade lengths, short refixations, and the total number of fixations. On the other hand, the word-specific features contained gaze duration on each word and the number of revisits on each word. Based on the same dataset as Smyrnakis et al., Asvestopoulou et al. (2019) developed a tool called DysLexML. The classifier with the highest accuracy on noise-free data is linear SVM, used on features selected by LASSO regression at λ1SE, which gave an accuracy of 87.87%, and up to 97%+ when using leave-one-out cross-validation. In recent years, Raatikainen et al. (2021) used a hybrid method consisting of an SVM classifier with random forest feature selection for dyslexia detection with data recorded from eye movement. The best-performing SVM model of their study scored an accuracy of 89.7%.

SUBJ	SCORE	n TEXTS	WPM	AGE	GENDER	DIAGNOSED
READERS WITH DYSLEXIA						
P23	1.00	2	200.0	33	F	16
P24	0.80	2	203.7	64	F	9
P25	0.82	4	142.0	20	F	16
P26	0.57	2	86.7	32	M	12
P27	0.71	4	137.4	53	M	48
P28	0.93	4	173.3	25	F	15
P29	0.73	3	143.3	25	F	21
P30	0.93	4	179.0	61	M	50
P31	0.75	2	61.9	20	M	15
P33	0.86	2	59.3	30	F	8
P34	0.62	2	107.4	56	F	9
P35	0.71	4	285.1	24	F	19
P36	0.40	2	58.5	23	F	11
P37	0.58	4	270.7	25	F	23
P38	0.75	2	115.5	30	M	29
P39	1.00	1	160.2	32	F	17
P40	0.92	4	173.3	29	M	7
P41	0.88	4	154.9	51	F	50
AVG	**0.78** (0.16)	**2.9** (1.1)	**150.7** (65.0)	**35.1** (14.7)	**67.7%F**	**20.8** (14.3)
READERS WITHOUT DYSLEXIA						
AVG	**0.81** (0.11)	**4.4** (1.5)	**276.8** (54.6)	**30.7** (10.8)	**78% F**	–

Table 1: Overview of readers with dyslexia included in the study. Average and standard deviations are in brackets. SCORE is the accuracy of the answers to the comprehension questions; DIAGNOSED refers to the age at which the participants were diagnosed with dyslexia. Aggregated data from the 18 readers without dyslexia from Hollenstein et al. (2022) for comparison.

3 Data Collection

Data acquisition follows Hollenstein et al. (2022), but the most important points are repeated here. The only procedural difference is the additional two reading tests administered to participants with dyslexia as described in §3.3.

3.1 Participant Selection

The participant selection for this study of natural reading is purposefully broad and follows the requirements for Hollenstein et al. (2022) from which we sample the typical readers. Prior to this, we excluded four participants from the non-dyslexic group from the analysis due to poor calibration or reported attention deficit disorder. The only difference to our participant sampling is that all dyslexic readers are officially diagnosed with dyslexia. There is no age limit and no required educational background but all participants are adults, and native speakers of Danish. All have normal vision or corrected-to-normal (glasses or contact lenses), but no readers included in the analysis had a known attention deficit disorder. All participants signed an informed consent and all digital data is pseudonymised. Due to the absence of an official dyslexia diagnosis, we discard the data from one subject for further analysis but include 18 readers in the dyslexic group. Participant statistics for all included dyslexic participants are presented in Table 1 with a summary of the 18 non-dyslexic participants for comparison.

3.2 Reading Materials

We used the same set of reading materials as Hollenstein et al. (2022) presented in the same way. They are 46 transcribed and proofread Danish speeches, accessed from the Danske Taler archive (https://dansketaler.dk). Table 2 shows an overview. The readability of each speech was calculated from a LIX score, which is based on the length of the words and sentences in a text (Björnsson, 1968). Each reader read a subset of the full dataset reported in n TEXTS in Table 1.

Reading Comprehension Questions To prevent mindless reading, comprehension questions were added to occur after approximately 20% of the paragraphs that contain more than 100 characters following Hollenstein et al. (2022). The average accuracy of the comprehension questions per participant can be seen in Table 1 in the SCORE

	MIN	MAX	MEAN	STD	TOTAL
SENTS PER DOC	37	134	92.4	29.4	1,849
TOKENS PER DOC	978	2,846	1,744.8	533.1	34,897
WORD TYPES PER DOC	391	1,056	603.6	159.4	7,361
LIX PER DOC	26.4	50.1	37.2	7.2	–
FREQUENCY PER DOC	0.68	0.79	0.74	0.03	–
SENT LEN IN TOKENS	1	119	10.8	15.9	–
TOKEN LEN IN CHARS	1	33	4.5	3.0	–

Table 2: Statistics on the 46 documents that comprise the reading material. TOTAL is the dataset total. LIX is the readability score. For typical readers, a text with a LIX score between 25 and 34 is considered easy, whereas a text scoring more than 55 is considered difficult and corresponds to an academic text. The frequency is measured by the proportion of words included in the 10,000 most common Danish words from `https://korpus.dsl.dk/resources/details/freq-lemmas.html` column.

3.3 Lexical Assessment

All participants with dyslexia performed two lexical assessment tests, which are used as a control test for the current study. Both tests are developed by the Centre of Reading Research, University of Copenhagen. The purpose of the tests is to have a comparable benchmark for a lexical assessment unrelated to the eye movements of the participants with dyslexia.

Nergård-Nilssen and Eklund (2018) found in their psychometric evaluation that a pseudohomophone test is of high reliability and that such a test incorporates evaluations that provide accurate discrimination of readers with dyslexia. Due to this finding, as well as the fact that the pseudohomophone task is used in the Danish dyslexia test, a pseudohomophone test was selected as one of the lexical assessment tests for the current study. For the sake of reliability and providing insightful findings on reading skills, a reading comprehension test was also used as a complementary lexical assessment test.

Reading Comprehension Test The original purpose of the reading comprehension test[5] is to provide easy access for adults to receive an informal evaluation of their reading skills, and to stress that more adults are seeking help with developing their reading skills (Jensen et al., 2014). It takes ten minutes to complete, making it relatively short, yet insightful. The tasks in the test consist of three variants of cloze tests, which are tests where the participants must select a missing word in a sentence, e.g., `It had been raining for some _____ [days, moments, countries]` (our translation).

As the reading task is an online self-assessment test that requires no log-in or external assistance, requirements, or access, the participants without dyslexia in the experiment were contacted after their participation in the eye-tracking experiment to voluntarily take the test at home to serve as a control group. Ten participants without dyslexia submitted their scores as a contribution to this experiment.

The aggregated results for both reader groups are presented in Table 3. We observe that readers with dyslexia generally have a lower score and a larger variance. A two-tailed t-test showed that this difference is significant ($p < 0.001$).

GROUP	n	MEAN	MIN	MAX
DYSLEXIC	18	3.5	0.7	5.2
NOT DYSLEXIC	10	5.7	4.4	7.1

Table 3: Reading task scores for participants of both reading groups. A score between 0–3.4 indicates that the reader may find many texts difficult and time-consuming to read, and a score between 3.5–3.9 indicates that the reader may find some texts difficult and/or time-consuming to read. A score over 4 indicates good reading skills.

Pseudohomophone Test The second linguistic assessment we conducted with the participants with dyslexia was a pseudohomophone [6] and was developed as a part of a diagnostic reading test for adults. The test encompasses 38 tasks where each task consists of four non-words, of which one of the words sounds like a real Danish word when pronounced. The difficulty of the 38 tasks increases gradually. The participants get five minutes to complete as many tasks as possible. Knowledge of the words of the test is required to perform it, but as the words are frequent, everyday words in Danish, it is assumed that native, adult readers are familiar with the words. Translated examples of the words are: cheese, eat, steps, factory, and help.

[5] Accessed from `https://selvtest.nu/`

[6] Accessed from `https://laes.hum.ku.dk/test/`

GROUP	n	ACC
NO READING DIFFICULTIES	72	66%
IN PROGRAMS FOR DYSLEXIC STUDENTS	46	23%
IN LITERACY READING PROGRAMS	167	31%
COPCO READERS WITH DYSLEXIA	18	33%

Table 4: Pseudohomophone test accuracies. The three top rows are standards from the official documentation of the test material for comparison.

The result is presented in Table 4 compared to standard scores from the documentation of the test[7]. We observe that the scores from the readers with dyslexia in the current study are on par with the standard scores of adults in literacy reading programs and higher than the standards for adults in programs for dyslexic readers. However, all quartile scores for our group of readers with dyslexia are about half compared to the standards for adults without reading difficulties.

3.4 Experiment Procedure

Eye movement data were collected with an infrared video-based EyeLink 1000 Plus eye tracker (SR Research) and follow Hollenstein et al. (2022). The experiment was designed with the SR Experiment Builder software. Data is recorded with a sampling rate of 1000Hz. Participants were seated at a distance of approximately 85 cm from a 27-inch monitor (display dimensions 590 x 335 mm, resolution 1920 x 1080 pixels). We recorded monocular eye-tracking data of the right eye. In a few cases of calibration difficulties, the left eye was tracked.

A 9-point calibration was performed at the beginning of the experiment. The calibration was validated after each block. Re-calibration was conducted if the quality was not good (worst point error $< 1.5°$, average error $< 1.0°$).Drift correction was performed after each trial, i.e. each screen of text. Minimum calibration quality measure of the recording ("good" calibration score, or "fair" in exceptionally difficult cases).

Experiment Protocol Participants read speeches in blocks of two speeches. The experiment was self-paced meaning there were no time restrictions. Thus, the participants read in their own pace for comprehension which is what we dub 'natural reading'. Between blocks, the participants could take a break. Each participant completed as many blocks as they were comfortable within one session. The order of the blocks and the order of the speeches within a block were randomized. Instructions were presented orally and on the computer screen before the experiment started. All participants first completed a practice round of reading a short speech with one comprehension question. The experiment duration was between 60 and 90 minutes.

Stimulus Presentation The text passages presented on each screen resembled the author's original division of the story into paragraphs as much as possible. Comprehension questions were presented on separate screens. The text was in a black, monospaced font (type: Consolas; size: 16pt) on a light-gray background (RGB: 248,248,248). The texts spanned max. 10 lines with triple line spacing. We used a 140 pixels margin at the top and bottom, and 200 pixels side margin for a screen resolution of 1920x1080.

4 Data Processing

4.1 Event Detection

This procedure also follows Hollenstein et al. (2022) closely. During data acquisition, the eye movement events are generated in real-time by the EyeLink eye tracker software during recording with a velocity- and acceleration-based saccade detection method. The algorithm defines a fixation event as any period that is not a saccade or a blink. Hence, the raw data consist of (x,y) gaze location coordinates for individual fixations.

We use the DataViewer software by SR Research to extract fixation events for all areas of interest. Areas of interest are automatically defined as rectangular boxes surrounding each text character on the screen, as shown in Figure 1. For later analysis, only fixations within the boundaries of each displayed character are extracted. Therefore, data points distinctly not associated with reading are excluded. We also set a minimum duration threshold of 100ms.

4.2 Feature Extraction

In the second step, we use custom Python code to map and aggregate character-level features to word-level features. These features cover the reading process from early lexical access to later syntactic integration. The selection of features is

[7] https://laes.hum.ku.dk/test/find_det_der_lyder_som_et_ord/standarder/

Figure 2: Correlation matrices showing correlations between all features recorded from readers with dyslexia (DR; left) and readers without dyslexia (NDR; right).

inspired by similar corpora in other languages (Siegelman et al., 2022; Hollenstein et al., 2018; Cop et al., 2017) as well as features known to show strong effects in eye movements from readers with dyslexia (Biscaldi et al., 1998; Pirozzolo and Rayner, 1979; Rayner, 1986). We extract the following eye-tracking features:

- nFIX: The total number of fixations on the current word.
- FFD: Duration of the first fixation of the current word.
- MFD: Mean duration of all fixations on the current word.
- TFD: Total fixation duration on the current word.
- FPD: first pass duration, The summed duration of all fixations on the current word prior to progressing out of the current word (left or right).
- GPT: go-past time, the sum duration of all fixations prior to progressing to the right of the current word, including regressions to previous words that originated from the current word.
- MSD: mean saccade duration, Mean duration of all saccades originating from the current word.
- PSV: peak saccade velocity, Maximum gaze velocity (in visual degrees per second) of all saccades originating from the current word.

The feature correlations for readers with and without dyslexia are shown in Figure 2. They illustrate that the correlation of the features is generally higher for readers without dyslexia compared to those with dyslexia. This may indicate that the data varies more among readers with dyslexia, suggesting that the reading pattern of the participants with dyslexia includes greater variability. The highest correlated features are those related to fixations, with the highest correlated pairs being first fixation duration and mean fixation duration, as well as total fixation duration and the number of fixation duration. A t-test analysis was performed to compare the features recorded from readers with and without dyslexia, revealing that all eight features show a significant difference between groups (p <0.0001).

5 Dyslexia Classification

We experiment with three types of classifiers using features on two different levels of aggregation; sentence-level and trial-level. A trial corresponds to the text presented on a single screen, roughly corresponding to paragraphs from the original text materials. For both levels of aggregation, the eye-tracking features of each word in a sentence or trial, respectively, are averaged to get a single vector of eight features for each sample. Further, we experiment with adding standard deviations (+STD) and max values (+MAX). Therefore, we

Experiment type	n samples Non-dyslexic	Dyslexic
Trial-level	5,147	4,144
Sentence-level	21,859	17,477

Table 5: Dataset size.

train classifiers, where each sample corresponds either to the eye-tracking information from a sentence or from a full trial. Dataset sizes are presented in Table 5. The data is split into 90% training data and 10% test data. We use an additional 10% of the training data as a validation split for the Long Short-Term Memory (LSTM). For all experiments, we randomly undersampled the non-dyslexic datasets for training, but not testing. We perform 5 runs taking different random samples from the data of readers without dyslexia and report the average performance.

SVM and Random Forest Classifiers The eye-tracking features are normalised with a min-max scaler that gives each instance a number between 0 and 1.We use a grid search to tune the hyper-parameters of both SVM (the best regularization parameter $C = 100$) and random forest (the best maximum depth=9, and the optimal number of estimators=200) in a 5-fold cross-validation setup on the full train set. The classifiers are implemented with the scikit-learn library for Python. The SVM uses a linear kernel. In addition to taking the mean feature values per word or trial (i.e., aggregating the eye-tracking features of all individual words), we also experiment with adding the standard deviations and maximum values of each feature.

LSTM Classifiers with Sequential Word Features We train a recurrent neural network optimized for sequential data, namely an LSTM. As LSTMs perform well with sequences and data consisting of large vocabularies and are effective in memorizing important information, it can be beneficial to dyslexia detection to predict the probability of class for a sentence, given the observed words. Therefore, the inputs for the LSTM network are the same eye-tracking features, but rather than aggregating on the full trial or sentence, each word is assigned a feature vector. The sequences were then padded to the maximum sentence or trial length, respectively. We use two LSTM layers, with 32 and 16 dimensions, respectively, and a dropout rate of 0.3 after the first layer. Finally, we use a sigmoid activation function for outputting the probabilities of each class. The models are trained with a batch size of 128, using a cross-entropy loss and a RMSprop optimizer with a learning rate of 0.001. We implement early stopping with a patience of 70 epochs on the maximum validation accuracy and save the best model. The model was implemented using Keras.

Model	Trial	Sentence
SVM	0.80 (0.018)	0.71 (0.004)
SVM + std	0.81 (0.010)	0.71 (0.006)
SVM + std + max	0.81 (0.014)	0.72 (0.007)
RF	0.83 (0.012)	0.72 (0.001)
RF + std	**0.85** (0.015)	0.72 (0.007)
RF + std + max	**0.85** (0.010)	**0.73** (0.006)
LSTM	0.82 (0.030)	0.71 (0.037)

Table 6: Average F1 score (standard deviation across five runs in brackets) for SVM, R(random)F(orest) and LSTM.

5.1 Results

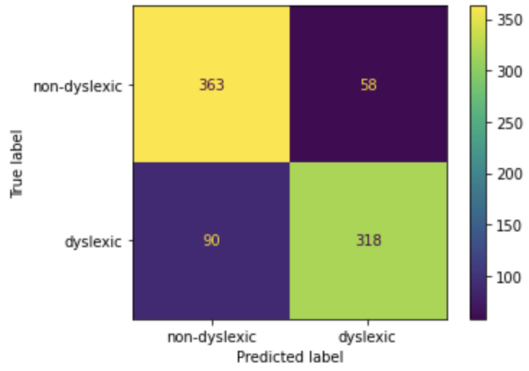
The trial-level and sentence-level results for the dyslexia classification task are presented in Table 6. We observe that trial-level classifiers achieve much higher results than sentence-level classifiers, which is to be expected since the latter includes reading data from fewer words. However, for the SVM and random forest, the features are aggregated. Hence there will be an upper limit of text length suitable for these methods. The random forest achieves the best results on both levels and a wider range of features (namely, including standard variation and maximum value features) yields higher scores. The LSTM model does not outperform the simpler and faster-to-train random forest models and shows a higher variance between runs.

5.1.1 Misclassifications

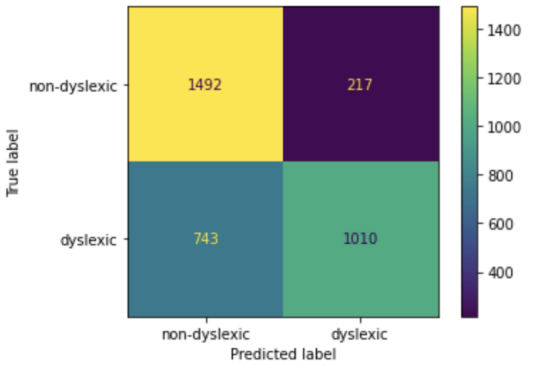
To further analyze these results, we look at the confusion matrix and misclassified participants from the best model, namely the random forest classifier including mean, standard deviation, and maximum value features. The confusion matrices in Figure 3 show that more mistakes are made classifying samples from readers with dyslexia than from readers without dyslexia. This is more apparent at sentence-level where the number of samples is substantially larger.

Furthermore, we hypothesize that the classifier struggles to correctly classify samples from read-

ers with dyslexia that have reading patterns comparable to readers without dyslexia. The samples that are misclassified most frequently belong mostly to the same group of participants, both at sentence-level and at trial-level. The most frequently misclassified samples from readers with dyslexia were P28, P35, P23, P40, and P37 (in descending order of the number of misclassifications). We correlate the number of misclassified samples for all participants with dyslexia with their demographic and lexical text information and find a significant correlation between misclassifications and words per minute ($\rho = 0.79, p < 0.001$) and between misclassifications and reading comprehension scores ($\rho = 0.71, p < 0.001$). However, the correlation between misclassifications and pseudohomophone test scores is minimal and not significant. This shows that samples from readers with dyslexia with higher reading speed and better reading comprehension are more likely to be misclassified since the features are more similar to readers without dyslexia.



(a) Trial-level



(b) Sentence-level

Figure 3: Confusion matrices for the best classifier, RF+SDT+MAX, for each experiment level.

6 Discussion & Conclusion

We presented a dataset of eye-tracking recordings from natural reading from adults with dyslexia, which complements the CopCo dataset of readers without dyslexia (Hollenstein et al., 2022). Additionally, to the best of our knowledge, we presented the first attempt to predict dyslexia from eye-tracking features using Danish as a target language. The best-performing classifier of the current study achieves an F1 score of 0.85, using a random forest classifier trained with a feature combination that includes the aggregation of means, standard deviations, and maximum values of eight eye-tracking features.

While the recorded eye-tracking features proved to reflect vital information about the reading mechanisms of the participants, there were a considerably high number of misclassifications of fast and skilled readers with dyslexia. This indicates that a fast reading speed is atypical for a reader with dyslexia. These results contribute to findings that the symptoms of dyslexia occur in varying degrees and thus underline the importance of developing a reliable assessment tool for dyslexia that can reduce the number of misclassifications.

Moreover, due to known comorbidities across reading disorders (Mayes et al., 2000) that can be reflected in eye movements (e.g., attention and autism spectrum disorders), as the dataset continues to grow, we will include these populations of readers in the data collection to learn to classify different subgroups readers correctly.

Precise criteria for dyslexia diagnosis remain difficult to standardise with the varying degrees of the symptoms and indicators of the disorder, which is why the condition deserves more attention. As eye-tracking recordings provide insightful information about cognitive processes in naturalistic tasks such as reading, they can be a beneficial tool for dyslexia prediction. Eye tracking can be a stepping stone to achieving more reliable screening methods for dyslexia.

Acknowledgments

Maria Barrett is supported by a research grant (34437) from VILLUM FONDEN.

References

Arwa Al-Edaily, Areej Al-Wabil, and Yousef Al-Ohali. 2013. Dyslexia explorer: A screening system for learning difficulties in the arabic language using eye tracking. In *International Conference on Human Factors in Computing and Informatics*, pages 831–834. Springer.

André Duncan Arceneaux. 2006. *It doesn't make any sense: Self and strategies among college students with learning disabilities*. Ph.D. thesis, University of Missouri–Columbia.

Thomais Asvestopoulou, Victoria Manousaki, Antonis Psistakis, Ioannis Smyrnakis, Vassilios Andreadakis, Ioannis M Aslanides, and Maria Papadopouli. 2019. Dyslexml: Screening tool for dyslexia using machine learning.

Mattias Benfatto, Gustaf Öqvist Seimyr, Jan Ygge, Tony Pansell, Agneta Rydberg, and Christer Jacobson. 2016. Screening for dyslexia using eye tracking during reading. *PloS one*, 11(12):e0165508.

Monica Biscaldi, Stefan Gezeck, and Volker Stuhr. 1998. Poor saccadic control correlates with dyslexia. *Neuropsychologia*, 36(11):1189–1202.

CH Björnsson. 1968. Läsbarhet, liber. *Stockholm, Sweden*.

Christoforos Christoforou, Argyro Fella, Paavo HT Leppänen, George K Georgiou, and Timothy C Papadopoulos. 2021. Fixation-related potentials in naming speed: A combined eeg and eye-tracking study on children with dyslexia. *Clinical Neurophysiology*, 132(11):2798–2807.

Uschi Cop, Nicolas Dirix, Denis Drieghe, and Wouter Duyck. 2017. Presenting geco: An eyetracking corpus of monolingual and bilingual sentence reading. *Behavior research methods*, 49(2):602–615.

Maria De Luca, Enrico Di Pace, Anna Judica, Donatella Spinelli, and Pierluigi Zoccolotti. 1999. Eye movement patterns in linguistic and non-linguistic tasks in developmental surface dyslexia. *Neuropsychologia*, 37(12):1407–1420.

Hazel Denhart. 2008. Deconstructing barriers: Perceptions of students labeled with learning disabilities in higher education. *Journal of learning disabilities*, 41(6):483–497.

Mark Eckert. 2004. Neuroanatomical markers for dyslexia: a review of dyslexia structural imaging studies. *The neuroscientist*, 10(4):362–371.

GF Eden, JF Stein, HM Wood, and FB Wood. 1994. Differences in eye movements and reading problems in dyslexic and normal children. *Vision research*, 34(10):1345–1358.

Burkhart Fischer and Heike Weber. 1990. Saccadic reaction times of dyslexic and age-matched normal subjects. *Perception*, 19(6):805–818.

Patrick Haller, Andreas Säuberli, Sarah Elisabeth Kiener, Jinger Pan, Ming Yan, and Lena Jäger. 2022. Eye-tracking based classification of mandarin chinese readers with and without dyslexia using neural sequence models. *arXiv preprint arXiv:2210.09819*.

John M Henderson. 2013. Eye movements. *The Oxford handbook of cognitive psychology*, pages 69–82.

Nora Hollenstein, Maria Barrett, and Marina Björnsdóttir. 2022. The copenhagen corpus of eye tracking recordings from natural reading of danish texts. *arXiv preprint arXiv:2204.13311*.

Nora Hollenstein, Jonathan Rotsztejn, Marius Troendle, Andreas Pedroni, Ce Zhang, and Nicolas Langer. 2018. Zuco, a simultaneous eeg and eye-tracking resource for natural sentence reading. *Scientific data*, 5(1):1–13.

Jukka Hyönä and Richard K Olson. 1995. Eye fixation patterns among dyslexic and normal readers: effects of word length and word frequency. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21(6):1430.

Katrine Lyskov Jensen, Anna Steenberg Gellert, and Carsten Elbro. 2014. Rapport om udvikling og afprøvning af selvtest af læsning – en selvtest af voksnes læsefærdigheder på nettet.

Holger Juul and Baldur Sigurdsson. 2005. Orthography as a handicap? a direct comparison of spelling acquisition in danish and icelandic. *Scandinavian Journal of Psychology*, 46(3):263–272.

Shahriar Kaisar. 2020. Developmental dyslexia detection using machine learning techniques: A survey. *ICT Express*, 6(3):181–184.

Joanna Lynne Kelm. 2016. *Adults' experiences of receiving a diagnosis of a learning disability*. Ph.D. thesis, University of British Columbia.

Shelley Young Kong. 2012. The emotional impact of being recently diagnosed with dyslexia from the perspective of chiropractic students. *Journal of Further and Higher Education*, 36(1):127–146.

Susan D Mayes, Susan L Calhoun, and Errin W Crowell. 2000. Learning disabilities and adhd: Overlapping spectrum disorders. *Journal of learning disabilities*, 33(5):417–424.

Trude Nergård-Nilssen and Kenneth Eklund. 2018. Evaluation of the psychometric properties of "the norwegian screening test for dyslexia". *Dyslexia*, 24(3):250–262.

Boris Nerušil, Jaroslav Polec, Juraj Škunda, and Juraj Kačur. 2021. Eye tracking based dyslexia detection using a holistic approach. *Scientific Reports*, 11(1):1–10.

George Th Pavlidis. 1981. Do eye movements hold the key to dyslexia? *Neuropsychologia*, 19(1):57–64.

Harshani Perera, Mohd Fairuz Shiratuddin, and Kok Wai Wong. 2018. Review of eeg-based pattern classification frameworks for dyslexia. *brain informatics*, 5(2):1–14.

Francis J Pirozzolo and Keith Rayner. 1979. The neural control of eye movements in acquired and developmental reading disorders. *Studies in neurolinguistics*, pages 97–123.

A Jothi Prabha and R Bhargavi. 2020. Predictive model for dyslexia from fixations and saccadic eye movement events. *Computer Methods and Programs in Biomedicine*, 195:105538.

Peter Raatikainen, Jarkko Hautala, Otto Loberg, Tommi Kärkkäinen, Paavo Leppänen, and Paavo Nieminen. 2021. Detection of developmental dyslexia with machine learning using eye movement data. *Array*, 12:100087.

Maria Rauschenberger, Luz Rello, Ricardo Baeza-Yates, Emilia Gomez, and Jeffrey P Bigham. 2017. Towards the prediction of dyslexia by a web-based game with musical elements. In *Proceedings of the 14th International Web for All Conference*, pages 1–4.

Keith Rayner. 1986. Eye movements and the perceptual span in beginning and skilled readers. *Journal of experimental child psychology*, 41(2):211–236.

Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372.

University of Copenhagen Centre for Reading Research, Danish School of Education, Aarhus University, The National Board of Social Services, Ministry of Children, and Education. 2020. *Vejledning til Ordblindetesten (version 8)*. Ministry of Children and Education.

Luz Rello and Miguel Ballesteros. 2015. Detecting readers with dyslexia using machine learning with eye tracking measures. In *Proceedings of the 12th International Web for All Conference*, pages 1–8.

CA Rubino and HA Minden. 1973. An analysis of eye-movements in children with a reading disability. *Cortex: A Journal Devoted to the Study of the Nervous System and Behavior*.

Gerd Schulte-Körne. 2010. The prevention, diagnosis, and treatment of dyslexia. *Deutsches Ärzteblatt International*, 107(41):718.

Noam Siegelman, Sascha Schroeder, Cengiz Acartürk, Hee-Don Ahn, Svetlana Alexeeva, Simona Amenta, Raymond Bertram, Rolando Bonandrini, Marc Brysbaert, Daria Chernova, et al. 2022. Expanding horizons of cross-linguistic research on reading: The multilingual eye-movement corpus (meco). *Behavior research methods*, pages 1–21.

Chris Singleton, Joanna Horne, and Fiona Simmons. 2009. Computerised screening for dyslexia in adults. *Journal of Research in Reading*, 32(1):137–152.

Ioannis Smyrnakis, Vassilios Andreadakis, Vassilios Selimis, Michail Kalaitzakis, Theodora Bachourou, Georgios Kaloutsakis, George D Kymionis, Stelios Smirnakis, and Ioannis M Aslanides. 2017. Radar: A novel fast-screening method for reading difficulties with special focus on dyslexia. *PloS one*, 12(8):e0182597.

Kathleen Tanner. 2009. Adult dyslexia and the 'conundrum of failure'. *Disability & Society*, 24(6):785–797.

Is Part-of-Speech Tagging a Solved Problem for Icelandic?

Örvar Kárason
Department of Computer Science
Reykjavik University
Iceland
orvark13@ru.is

Hrafn Loftsson
Department of Computer Science
Reykjavik University
Iceland
hrafn@ru.is

Abstract

We train and evaluate four Part-of-Speech tagging models for Icelandic. Three are older models that obtained the highest accuracy for Icelandic when they were introduced. The fourth model is of a type that currently reaches state-of-the-art accuracy. We use the most recent version of the MIM-GOLD training/testing corpus, its newest tagset, and augmentation data to obtain results that are comparable between the various models. We examine the accuracy improvements with each model and analyse the errors produced by our transformer model, which is based on a previously published ConvBERT model. For the set of errors that all the models make, and for which they predict the same tag, we extract a random subset for manual inspection. Extrapolating from this subset, we obtain a lower bound estimate on annotation errors in the corpus as well as on some unsolvable tagging errors. We argue that further tagging accuracy gains for Icelandic can still be obtained by fixing the errors in MIM-GOLD and, furthermore, that it should still be possible to squeeze out some small gains from our transformer model.

1 Introduction

Part-of-Speech (POS) tagging is a sequential labelling task in which each token, i.e., words, symbols, and punctuation in running text is assigned a morphosyntactic tag. It is an important step for many Natural Language Processing applications. A token is ambiguous when it has more than one possible tag. The source of ambiguity is polysemy in the form of homographs from the same word class, from different word classes, and also within the declension paradigms of the same word. The task, therefore, entails examining the token itself and its context for clues for predicting the correct tag. For the last mentioned type of ambiguity, which is prevalent in Icelandic, it is necessary to find another unambiguous token in the context that the target token shows agreement with and use it to determine the correct target tag.

Over the last two decades, steady progress has been made in POS tagging for Icelandic. Various taggers have been presented throughout this period that improved on previous state-of-the-art (SOTA) methods (Rögnvaldsson et al., 2002; Helgadóttir, 2005; Loftsson, 2008; Dredze and Wallenberg, 2008; Loftsson et al., 2009, 2011; Loftsson and Östling, 2013; Steingrímsson et al., 2019; Snæbjarnarson et al., 2022; Daðason and Loftsson, 2022; Jónsson and Loftsson, 2022).

Work on Icelandic corpora has also progressed. Existing corpora have undergone error correction phases (Barkarson et al., 2021), and, in some cases, been expanded with new data (Barkarson et al., 2022). A new larger gold standard corpus for POS tagging, *MIM-GOLD* (Loftsson et al., 2010), was created to replace the older standard, the *Icelandic Frequency Dictionary* (IFD, Pind et al. 1991), and multiple alterations have been made to the fine-grained Icelandic tagset (Steingrímsson et al., 2018; Barkarson et al., 2021).

All this variability over the years means that previously reported results for POS taggers are not easily comparable. Thus, we train and test four data-driven taggers that have been employed for Icelandic (see Section 3), using the latest version of MIM-GOLD and its underlying tagset, as well as the latest versions of augmentation data (see Section 2). We obtain SOTA tagging accuracy by training and fine-tuning a ConvBERT-base model in a slightly different manner than previously reported by Daðason and Loftsson (2022) (see Section 3).

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 71–79

With the latest tagging method based on the transformer model finally reaching above 97% per-token accuracy for Icelandic (Jónsson and Loftsson, 2022; Snæbjarnarson et al., 2022; Daðason and Loftsson, 2022), the generally believed limit of inter-annotator agreement (Manning, 2011), we might ask ourselves if POS tagging is now a solved problem for Icelandic. Indeed, our evaluation results show that the tagging accuracy of our ConvBERT-base model is close to 98% (see Table 3). A large portion of the remaining errors can be explained by 1) a lack of context information to make the correct prediction, and 2) annotation errors or other faults in the training/testing corpus itself. Addressing the latter should give further gains. Furthermore, some small additional gains could be squeezed out of the transformer model, by using a larger model and pre-training it on more data. When this is done, we may be able to argue that POS tagging is a solved problem for Icelandic.

The rest of this paper is structured as follows. In Sections 2 and 3, we describe the data and the models, respectively, used in our experiments. We present the evaluation results in Section 4, and detailed error analysis in Section 5. Finally, we conclude in Section 6.

2 Data

In this section, we describe the data and the tagset used in our work.

2.1 Corpus

The MIM-GOLD corpus is a curated subset of the MIM corpus (Helgadóttir et al., 2012) and was semi-automatically tagged using a combination of taggers (Loftsson et al., 2010). Version 21.05 of the corpus contains 1 million running words from 13 different text types, of which about half originate from newspapers and books (see Table 1). All versions of MIM-GOLD include the same 10-fold splits for use in cross-validation.[1]

MIM-GOLD was created to replace the IFD as the gold standard for POS tagging of Icelandic texts. The IFD corpus was sourced from books published in the eighties and has a clear literary and standardized language slant. Steingrímsson et al. (2019) reported a 1.11 percentage point (pp) lower per-token accuracy for MIM-GOLD compared to the IFD.

Text type	% of all
Newspaper *Morgunblaðið*	24.9
Books	23.5
Blogs	13.4
Newspaper *Fréttablaðið*	9.4
The Icelandic Web of Science	9.1
Websites	6.5
Laws	4.1
School essays	3.4
Written-to-be-spoken	1.9
Adjudications	1.3
Radio news scripts	1.1
Web media	0.8
E-mails	0.5
Total	100.0

Table 1: Information about the various text types in MIM-GOLD, adapted from Loftsson et al. (2010).

2.2 Morphological lexicon

Version 22.09 of the Database of Modern Icelandic Inflection (DMII) (Bjarnadóttir, 2012), which is now a part of the Database of Icelandic Morphology (Bjarnadóttir et al., 2019), contains 6.9 million inflectional forms and about 330 thousand declension paradigms.[2] Though the database cannot be used directly to train a POS tagger, as there is no context or distributional information for the word forms, it has been used to augment taggers during training and help with tagging unknown words (words not seen during training) (Loftsson et al., 2011; Steingrímsson et al., 2019).

2.3 Pre-training corpus

The Icelandic Gigaword Corpus (IGC), which includes text sources from multiple varied domains, has been expanded annually since 2018 (Barkarson et al., 2022). The motivation for constructing the IGC was, *inter alia*, to make the development of large Icelandic language models possible (Steingrímsson et al., 2018). The 2021 version used in our work contains about 1.8 billion tokens.[3]

[1] Version 21.05 is available at `http://hdl.handle.net/20.500.12537/114`

[2] `https://bin.arnastofnun.is/DMII/LTdata/`

[3] Version 2021 is available at `http://hdl.handle.net/20.500.12537/192`

2.4 Tagset

The MIM-GOLD tagset v. 2 is the fourth iteration of the fine-grained tagset that is exclusively used for modern Icelandic and has its origin in the IFD. The tagset consists of 571 possible tags, of which 557 occur in MIM-GOLD.

The tags are morphosyntactic encodings consisting of one to six characters, each denoting some feature. The first character denotes the *lexical category* and is, in some cases, followed by a sub-category character. For each category, a fixed number of additional feature characters follow, e.g., *gender*, *number* and *case* for nouns; *degree* and *declension* for adjectives; and *voice*, *mood* and *tense* for verbs. To illustrate, consider the word form *konan* ('the woman'). The corresponding tag is *nveng*, denoting noun (*n*), feminine (*v*), singular (*e*), nominative (*n*) case, and definite suffixed article (*g*).

3 Models

In this section, we describe the four data-driven POS tagging models we trained and evaluated:

- **TriTagger** (Loftsson et al., 2009) is a reimplementation of TnT (Brants, 2000), a second order (trigram) Hidden Markov model. The probabilities of the model are estimated from a training corpus using maximum likelihood estimation. Assignments of POS tags to tokens is found by optimising the product of lexical probabilities ($p(w_i|t_j)$) and contextual probabilities ($p(t_i|t_{i-1}, t_{i-2})$) (where w_i and t_i are the i^{th} word and tag, respectively).

  When work on creating a tagger for Icelandic started at the turn of the century, five existing data-driven taggers were tested on the IFD corpus (Helgadóttir, 2005). TnT obtained the highest accuracy and has often been included for comparison in subsequent work.

- **IceStagger** (Loftsson and Östling, 2013) is an averaged perceptron model (Collins, 2002), an early and simple version of a neural network.[4] It learns binary feature functions from predefined templates. The templates are hand-crafted and can reference adjacent words, previous tags, and various custom matching functions applied to them. The templates, intended to capture dependencies specific to Icelandic, were developed against the IFD. During training, the algorithm learns which feature functions are good indicators of the assigned tag, given the context available to the templates. It does that by adjusting the weight associated with the feature function. The highest-scoring tag sequence is approximated using beam search. Both IceStagger and TriTagger use data from the DMII to help with guessing the tags for unknown tokens.

- **ABLTagger v. 1** (Steingrímsson et al., 2019; Jónsson and Loftsson, 2022) is based on a bidirectional long short-term memory (Bi-LSTM) model.[5] That model is an extension of LSTMs (Hochreiter and Schmidhuber, 1997) that can be employed when the input is the whole sequence. Two LSTMs are trained on the input, with the second traversing it in reverse (Graves and Schmidhuber, 2005). The input for ABLTagger consists of both word and character embeddings. The model is augmented with n-hot vectors created from all the potential lexical features of the word forms from the DMII. ABLTagger was developed against the IFD but was the first tagger to be applied to MIM-GOLD.

- **ConvBERT** (Jiang et al., 2020) is an improved version of the BERT model (Vaswani et al., 2017; Devlin et al., 2019) that is more efficient and accurate. We used an existing ConvBERT-base model pre-trained on the IGC by Dađason and Loftsson (2022)[6] and fine-tuned it for tagging on MIM-GOLD. This is a standard pre-trained transformer model with two changes: the embeddings of the first and last subwords are concatenated (`first+last` subword pooling) to generate the token representations (Schuster and Nakajima, 2012), and we continued the pre-training of the ConvBERT-base model using the training data of each fold from MIM-GOLD for three epochs before fine-tuning it for tagging for 10 epochs with the same data. Each modification gave a 0.07 pp

[4] IceStagger and TriTagger are included in the IceNLP toolkit (Loftsson and Rögnvaldsson, 2007): `https://github.com/hrafnl/icenlp`

[5] ABLTagger v. 1 is available at `https://hdl.handle.net/20.500.12537/53`

[6] `https://huggingface.co/jonfd/convbert-base-igc-is`

	Token acc.	Sent. acc.
TriTagger	91.01%	35.58%
IceStagger	92.72%	42.74%
ABLTagger v1	94.56%	49.11%
ConvBERT-base	**97.79%**	**73.43%**

Table 2: Token and sentence tagging accuracy for the four models.

improvement in accuracy; i.e. 0.14 pp in total.[7]

4 Results

We evaluated the four models by applying 10-fold cross-validation (CV) using the standard splits in MIM-GOLD (see Section 2). The results are shown in Table 2. The transformer model, ConvBERT-base, obtains 6.78 pp higher accuracy than the HMM model (TriTagger), which is equivalent to a 75.42% reduction in errors!

The increase in sentence accuracy, which is often overlooked, is also very impressive. It has more than doubled and now close to $\frac{3}{4}$ of the sentences are correct. Sentences come in different lengths, ranging from a single token up to 1,334 tokens in MIM-GOLD, and increased length can result in increased complexity. Figure 1 shows the length distribution of sentences with no errors. The figure shows both general accuracy gains as well as an improvement in handling longer sentences.

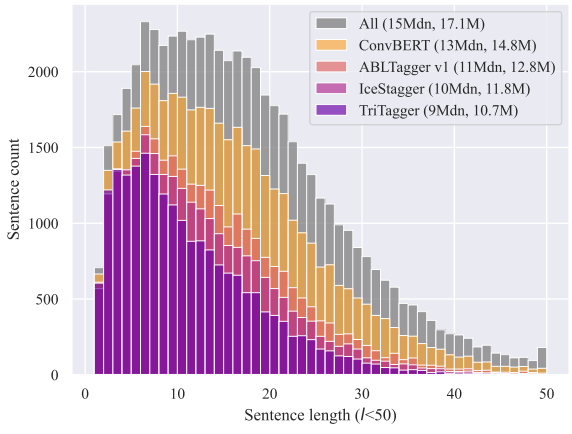


Figure 1: Distributions of correctly tagged sentences. The legend shows each set's median (Mdn) and mean (M).

[7]See `https://github.com/orvark13/postr/` for training and evaluation scripts, as well as fine-tuned models.

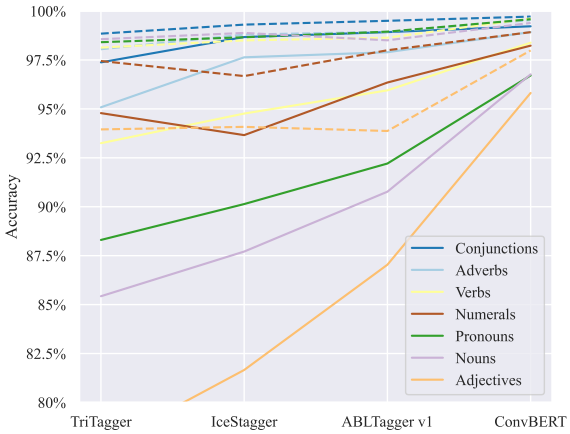


Figure 2: The accuracy improvements between the models for the more frequent lexical categories. Solid lines are the per-token accuracy for all tags in that category, and dashed lines are the lexical class accuracy, i.e., the tag category is correct but there is some error in the predicted features. Errors within the categories diminish as those lines converge.

4.1 Accuracy improvements

TriTagger and IceStagger are limited to a three-token window and they need frequency information of tokens to learn from. As is to be expected, IceStagger gains accuracy according to the feature templates pre-defined for it. ABLTagger's improvements come from the BiLSTM's context window being the whole sentence and it, thereby, being able to detect long-range dependencies. Its ability to see within the token by means of the character embeddings helps it handle tokens not seen during training. Augmenting the model with data from DMII also helps with unknown words.

The source of improvement for the transformer model is mainly threefold. First, the attention mechanism aids it in selecting the right dependencies (e.g., when there is more than one option), and it is detecting longer long-range dependencies than the BiLSTM model. We see this from the examination of the predictions and it is also indicated by the model's success with longer sentences as is evident in the shape of its distribution in Figure 1. Secondly, the model is often able to discern the different semantic senses of ambiguous tokens. We assume this stems from the contextual word embeddings in the large pre-trained ConvBERT language model. Finally, it benefits from all the language sense from the IGC infused in the

POS Transformer Model	Accuracy
IceBERT-IGC [1]	97.37%
ConvBERT-base [1]	97.75%
Our ConvBERT-base	**97.79%**
Excluding *x* and *e* tags	
IceBERT-IGC, multi-label [2]	**98.27%**
Our ConvBERT-base	98.14%
9-fold CV, excluding *x* and *e* errors	
DMS, ELECTRA-base [3]	97.84%
Our ConvBERT-base	**98.00%**

Table 3: Accuracy results for different POS transformer models pre-trained on IGC and the accuracy of our transformer model when fine-tuned and evaluated in a comparable manner. [1] were reported in Dađason and Loftsson (2022), [2] in Snæbjarnarson et al. (2022), and [3] in Jónsson and Loftsson (2022).

language model during pre-training.

Figure 2 shows the accuracy improvements of the models for the more frequent lexical categories.

4.2 Transformer models and SOTA

In Table 3, we show previously reported results for transformer models pre-trained on the IGC, and the results of our transformer, a ConvBERT-base model trained and fine-tuned slightly differently compared to Dađason and Loftsson (2022) (see Section 3), evaluated in the same manner for comparison. Two of the papers cited in the table report results excluding the *x* and *e* tags, either both during training and evaluation or only during evaluation. These tags are used for unanalysed tokens and foreign words, respectively, and have the lowest category accuracies, the reasons for which will become apparent in Section 5. Not counting tagging errors for these two tags increases reported accuracy by 0.21 pp for our model. Excluding those tags from training, by fixing their weights to zero, increases the reported accuracy by a further 0.14 pp, because, in this case, the model is no longer able to assign these two tags erroneously to tokens.

The current SOTA is a *multi-label* model based on IceBERT-large[8] (Snæbjarnarson et al., 2022). Multi-label classification means that the tags are split into individual features, e.g., *lexical category*, *tense*, *gender*, *number*, and the model is trained to predict each separately. Treating composite tags as multiple labels has been shown to improve POS tagging accuracy, especially when training data is scarce (Tkachenko and Sirts, 2018). Combining the predictions back into tags is dependent on knowledge about the composition of the tags. The results presented in Table 3 show that our ConvBERT-base model obtains SOTA results for single-label models applied to Icelandic.

5 Error analysis

In this section, we, first, present an analysis of the most frequent errors, and, second, the results of our analysis of the different sources of errors.

5.1 Most frequent errors

Table 4 shows the most frequent errors made by our transformer model. The list for the BiLSTM model is very similar, but with about double the accuracy degradation. The 12 most frequent errors are in fact six pairs of tags where the confusion between each pair occurs in either direction.

The most frequent confusion is *n—s→e* (and *e→n—s*), or between foreign proper names and foreign words.[9] More than half, 0.04 pp for both error types, are due to words not seen during training. According to the MIM-GOLD tagging guidelines, compound foreign names should have the first word tagged as a foreign proper name (*n—s*), and then the rest of the name tagged as foreign words (*e*), except for names of persons and places that should have all parts tagged as foreign proper names (*n—s*). The tag *n—s* is also used for abbreviations of foreign proper names, e.g., *BBC*. There are also some special cases that deviate from these rules (Barkarson et al., 2021). A significant portion of these tagging errors is indeed caused by annotation errors in the corpus (mostly *n—s→e*), as well as the fact that the application of the rules requires world knowledge that the models of course lack.

Confusion between adverbs and prepositions (which are annotated in MIM-GOLD as adverbs that govern case), i.e., *aa→af* (and *af→aa*) are the next most frequent errors. Some of these tagging errors are due to cases where there is a clause between the preposition and the object, or where the

[8] IceBERT is based on a RoBERTa model (Liu et al., 2019).

[9] We denote a tagging error with *a→b* where *a* is the predicted tag and *b* is the gold tag. The tag *n—s* stands for a proper noun without markings for gender, number, or case.

No.	Predicted tag → gold tag	Degradation in pp
1.	*n—s* → *e*	0.07
2.	*e* → *n—s*	0.07
3.	*af* → *aa*	0.05
4.	*aa* → *af*	0.05
5.	*nheo* → *nhfo*	0.03
6.	*fpheþ* → *faheþ*	0.03
7.	*nveþ* → *nveo*	0.03
8.	*nhfo* → *nheo*	0.02
9.	*nveo* → *nveþ*	0.02
10.	*ct* → *c*	0.02
11.	*c* → *ct*	0.02
12.	*faheþ* → *fpheþ*	0.02

Table 4: The 12 most frequent tagging errors our transformer model makes. The rightmost column shows accuracy degradation in percentage points for each error type.

object has been moved to the front of the sentence. There also seem to be a fair number of annotation errors associated with this confusion between adverbs and prepositions.

A confusion between personal and demonstrative pronouns, *fpheþ*→*faheþ* (and *faheþ*→*fpheþ*), is caused by the antecedent being out of context or being a whole clause. Understanding the clause is often necessary to make the distinction. These are all the same word form, *því* ('it' or 'this, that'). For *því/fpheþ*→*faheþ*, we see some improvement in accuracy with the transformer model over the other models, but for *því/faheþ*→*fpheþ*, we notice the only case of lower accuracy for the transformer model compared to the others. The tags here are for neuter (*h*) singular (*e*) in the dative case (*þ*). There are identical confusions for the accusative and genitive cases, but those tokens are not as frequent.

The *c*→*ct* (and *ct*→*c*) errors are comparative conjunctions being marked as relativizers (a subordinating conjunction indicating a relative clause) and vice versa. Except for a few antiquated uses of *er*, these cases are all the word form *sem* ('as' or 'who, whom, that, which'). The conjunction *sem* subsumed *er*'s role as a relativizer in Old Icelandic. This language change was feasible due to their syntactic structures being identical (Kemmer, 1984). Semantically their function is similar, as one complements and the other modifies a noun phrase with the following clause. The difference is this role of the relation. Therefore, the remaining tagging errors for *sem* are caused by a lack of syntactic and contextual information to make the correct prediction. Indeed, Loftsson et al. (2009) suggested that two tag categories be merged.

The errors *nheo*→*nhfo* (and *nhfo*→*nheo*), are confusions between the singular (*e*) and plural (*f*) forms of neuter nouns (*nh...*). When this error occurs, the context is usually not enough to determine the correct number. A wider context, previous sentences, or general knowledge is needed, and might even not be enough. Finally, *nveþ*→*nveo* (and *nveo*→*nveþ*) are confusions between the dative (*þ*) and accusative (*o*) cases of feminine nouns (*nv...*). The word that governs the case needs to be in the context, if it is omitted the distinction cannot be made. Moreover, if it can govern both cases, the required semantic information is unavailable.

One other group of errors should be mentioned, $* \rightarrow x$, where $*$ is any tag and the *x* tag denotes *unanalysed* tokens. This error is obscured because the predictions are distributed over many tags. These are tokens that contain spelling mistakes or constitute grammar errors and are the majority of the 2,777 tokens in the *unanalysed* tag category. Of the four models, the transformer does best with this tag category but is only predicting 58% correctly. Without changing how the spelling mistakes are annotated in MIM-GOLD or simply excluding sentences containing them, this will continue to be a source of about 0.12 pp accuracy degradation. As the corpus also contains tokens with such mistakes that are not annotated as *unanalysed* it would be in line with current practice to look to the intended meaning of these tokens and tag them accordingly.

5.2 Sources of errors

Manning (2011) discusses the generally perceived 97% token accuracy upper limit for POS tagging. At that time, those accuracy numbers had been reached for English, but Icelandic, a morphologically richer language with a very fine-grained tagset, had a long way to go. Rögnvaldsson et al. (2002) had earlier suggested 98% as the highest possibly achievable goal for Icelandic, because of inter-annotator disagreement. Manning reasons that the disagreement might actually be higher but says it is mitigated with annotator guidelines and

adjusting tag categories. Besides disagreement, subjectivity in annotation and the possibility of more than one right choice make up what Plank (2022) calls human label variation.

Manning samples errors the Stanford POS Tagger (Toutanova et al., 2003) makes when applied to a portion of the Penn Treebank corpus. He analyses the errors to try to understand if and how tagging accuracy could be further improved. He finds that the largest opportunity for gains is in improving the linguistic resources used to train the tagger. Before the initial release of MIM-GOLD, Steingrímsson et al. (2015) carried out an identical analysis on errors in both the IFD and MIM-GOLD when tagged with IceStagger. Their findings concurred with Manning's. We performed a similar analysis, though with a less detailed classification of the errors.

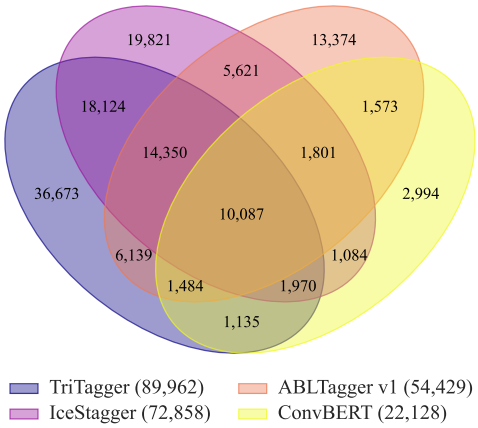


Figure 3: Venn diagram showing how prediction errors are shared between the four models.

Of the 1,000,218 tokens in MIM-GOLD, our transformer model makes 22,128 tagging errors. For 10,087 of these tokens, the three other taggers also make errors (see Figure 3), and for 5,526 of them, all four taggers agree on the predicted tag. From this set of errors, we drew a random sample of 500 for analysis. In this sample, we discovered 166 annotation errors, i.e., incorrect gold tags. For 150 of them, the taggers predicted the correct tag. Extrapolating to the superset gives us 1,658 tagging errors caused by gold errors (≈0.16 pp). We also found 87 cases where the prediction error was obviously caused by there being insufficient context information (≈0.09 pp), and 18 cases where it was likely caused by a spelling or grammar mistake (≈0.02 pp). The last error class (spelling or grammar mistakes) is aggravated by the use of the *unanalysed* tag (*x*) for such mistakes in the corpus. Table 5 shows the accuracy degradation for each of these error classes. Though we cannot draw conclusions from these findings about the frequency of these errors in the whole set of 22,128 errors, it is safe to assume these are the lower bounds of these error categories.

Error class	pp
Annotation errors	0.16
Insufficient context	0.09
Spelling or grammar mistakes	0.02
Unexplained	0.25
Total	0.52

Table 5: Estimated accuracy degradation in percentage points caused by each class in the set of prediction errors that all four taggers agree on.

6 Conclusions and Future Work

For Icelandic POS tagging, we have reached a point where individual error categories no longer stand out and annotation errors in the corpus are more pronounced, as well as inconsistencies stemming from human label variation.

Clear annotation errors can be corrected in the corpus, and the tagging guidelines and tag categories can be refined to remove some of the inconsistencies. Further gains can as well be squeezed out of the transformer model by using a larger model, i.e., ConvBERT-large instead of ConvBERT-base, increasing the vocabulary size, training it on the 2022 version of IGC that adds 549 million tokens, and fine-tuning the hyperparameters for the tagging model. Yet, on top of the annotator disagreement, there will always be errors because of a lack of information in the context, as well as the scarcity of examples to learn from for the long tail of infrequent tags.

For MIM-GOLD, that unsolvable part of the tagging errors seems to amount to less than 2 pp. Therefore, with a little more work, we should be able to confidently pass that 98% accuracy goal (when training and evaluating using the whole tagset) envisioned twenty years ago. A good starting point would be to search for and fix those estimated 1,658 annotation errors in MIM-GOLD, which are a subset of the tagging errors that all four models agree on.

To conclude, POS tagging for Icelandic is very close to being solved!

References

Starkaður Barkarson, Steinþór Steingrímsson, and Hildur Hafsteinsdóttir. 2022. Evolving Large Text Corpora: Four Versions of the Icelandic Gigaword Corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2371–2381, Marseille, France. European Language Resources Association.

Starkaður Barkarson, Þórdís Dröfn Andrésdóttir, Hildur Hafsteinsdóttir, Árni Davíð Magnússon, Kristján Rúnarsson, Steinþór Steingrímsson, Haukur Páll Jónsson, Hrafn Loftsson, Einar Freyr Sigurðsson, Eiríkur Rögnvaldsson, and Sigrún Helgadóttir. 2021. MIM-GOLD. Release notes with version 21.05.

Kristín Bjarnadóttir, Kristín Ingibjörg Hlynsdóttir, and Steinþór Steingrímsson. 2019. DIM: The Database of Icelandic Morphology. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 146–154, Turku, Finland. Linköping University Electronic Press.

Kristín Bjarnadóttir. 2012. The Database of Modern Icelandic Inflection. In *Proceedings of SaLTMiL-AfLaT Workshop on Language technology for normalisation of less-resourced languages*, LREC 2012, Istanbul, Turkey.

Thorsten Brants. 2000. TnT: A statistical part-of-speech tagger. *Applied Natural Language Processing Conference (ANLP)*, pages 224–231.

Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 1–8. Association for Computational Linguistics.

Jón Friðrik Daðason and Hrafn Loftsson. 2022. Pretraining and Evaluating Transformer-based Language Models for Icelandic. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7386–7391, Marseille, France. European Language Resources Association.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mark Dredze and Joel Wallenberg. 2008. Icelandic Data Driven Part of Speech Tagging. In *Proceedings of the* 46^{th} *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, ACL-HLT, Columbus, OH, USA.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *Neural Networks*, 18(5-6):602–610.

Sigrún Helgadóttir. 2005. Testing data-driven learning algorithms for PoS tagging of Icelandic. In H. Holmboe, editor, *Nordisk Sprogteknologi 2004*. Museum Tusculanums Forlag, Copenhagen.

Sigrún Helgadóttir, Ásta Svavarsdóttir, Eiríkur Rögnvaldsson, Kristín Bjarnadóttir, and Hrafn Loftsson. 2012. The Tagged Icelandic Corpus (MÍM). In *Proceedings of SaLTMiL-AfLaT Workshop on Language technology for normalisation of less-resourced languages*, LREC 2012, Istanbul, Turkey.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. 2020. ConvBERT: Improving BERT with Span-based Dynamic Convolution. In *Advances in Neural Information Processing Systems*, volume 33, pages 12837–12848. Curran Associates, Inc.

Haukur Jónsson and Hrafn Loftsson. 2022. DMS: A System for Delivering Dynamic Multitask NLP Tools. In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence - Volume 1: NLPinAI,*, pages 504–510. INSTICC, SciTePress.

Suzanne Kemmer. 1984. From Comparative to Relativizer: The case of Iceland Sem. *Annual Meeting of the Berkeley Linguistics Society*, 10:296–306.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Hrafn Loftsson. 2008. Tagging Icelandic text: A linguistic rule-based approach. *Nordic Journal of Linguistics*, 31(1):47–72.

Hrafn Loftsson, Sigrún Helgadóttir, and Eiríkur Rögnvaldsson. 2011. Using a Morphological Database to Increase the Accuracy in POS Tagging. In *Proceedings of Recent Advances in Natural Language Processing*, RANLP 2011, Hissar, Bulgaria.

Hrafn Loftsson, Ida Kramarczyk, Sigrún Helgadóttir, and Eiríkur Rögnvaldsson. 2009. Improving the PoS tagging accuracy of Icelandic text. In *Proceedings of the 17th Nordic Conference on Computational Linguistics (NODALIDA 2009)*, Odense, Denmark. Northern European Association for Language Technology (NEALT).

Hrafn Loftsson and Robert Östling. 2013. Tagging a Morphologically Complex Language Using an Averaged Perceptron Tagger: The Case of Icelandic. In

Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013), pages 105–119, Oslo, Norway. Linköping University Electronic Press, Sweden.

Hrafn Loftsson and Eiríkur Rögnvaldsson. 2007. IceNLP: A natural language processing toolkit for Icelandic. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 1, pages 1533–1536.

Hrafn Loftsson, Jökull H. Yngvason, Sigrún Helgadóttir, and Eiríkur Rögnvaldsson. 2010. Developing a PoS-tagged corpus using existing tools. In *Proceedings of 7th SaLTMiL Workshop on Creation and Use of Basic Lexical Resources for Less-Resourced Languages*, LREC 2010, Valetta, Malta.

Christopher D. Manning. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In Alexander Gelbukh, editor, *Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, volume 6608 of *Lecture Notes in Computer Science*, pages 171–189. Springer.

Jörgen Pind, Friðrik Magnússon, and Stefán Briem. 1991. *Íslensk orðtíðnibók [Icelandic frequency dictionary]*. Orðabók Háskólans, Reykjavik.

Barbara Plank. 2022. The "Problem" of Human Label Variation: On Ground Truth in Data, Modeling and Evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi. Association for Computational Linguistics.

Eiríkur Rögnvaldsson, Auður Rögnvaldsdóttir, Kristín Bjarnadóttir, and Sigrún Helgadóttir. 2002. Vélræn málfræðigreining með námfúsum markara [Automatic language analysis using a transformation-based tagger]. *Orð og tunga, Reykjavik*, 6:1–9.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *International Conference on Acoustics, Speech and Signal Processing*, pages 5149–5152.

Vésteinn Snæbjarnarson, Haukur Barri Símonarson, Pétur Orri Ragnarsson, Svanhvít Lilja Ingólfsdóttir, Haukur Jónsson, Vilhjalmur Thorsteinsson, and Hafsteinn Einarsson. 2022. A Warm Start and a Clean Crawled Corpus – A Recipe for Good Language Models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4356–4366, Marseille, France. European Language Resources Association.

Steinþór Steingrímsson, Sigrún Helgadóttir, and Eiríkur Rögnvaldsson. 2015. Analysing inconsistencies and errors in PoS tagging in two Icelandic gold standards. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 287–291, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Steinþór Steingrímsson, Sigrún Helgadóttir, Eiríkur Rögnvaldsson, Starkaður Barkarson, and Jón Guðnason. 2018. Risamálheild: A Very Large Icelandic Text Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Steinþór Steingrímsson, Örvar Kárason, and Hrafn Loftsson. 2019. Augmenting a BiLSTM Tagger with a Morphological Lexicon and a Lexical Category Identification Step. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1161–1168, Varna, Bulgaria.

Alexander Tkachenko and Kairit Sirts. 2018. Modeling Composite Labels for Neural Morphological Tagging. In *Conference on Computational Natural Language Learning*.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 252–259.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

MULTI-CROSSRE
A Multi-Lingual Multi-Domain Dataset for Relation Extraction

Elisa Bassignana **Filip Ginter** **Sampo Pyysalo**
Rob van der Goot **Barbara Plank**
Department of Computer Science, IT University of Copenhagen, Denmark
TurkuNLP, Department of Computing, University of Turku, Finland
MaiNLP, Center for Information and Language Processing, LMU Munich, Germany
elba@itu.dk figint@utu.fi

Abstract

Most research in Relation Extraction (RE) involves the English language, mainly due to the lack of multi-lingual resources. We propose MULTI-CROSSRE, the broadest multi-lingual dataset for RE, including 26 languages in addition to English, and covering six text domains. MULTI-CROSSRE is a machine translated version of CrossRE (Bassignana and Plank, 2022a), with a sub-portion including more than 200 sentences in seven diverse languages checked by native speakers. We run a baseline model over the 26 new datasets and—as sanity check—over the 26 back-translations to English. Results on the back-translated data are consistent with the ones on the original English CrossRE, indicating high quality of the translation and the resulting dataset.

1 Introduction

Binary Relation Extraction (RE) is a sub-field of Information Extraction specifically aiming at the extraction of triplets from text describing the semantic connection between two entities. The task gained a lot of attention in recent years, and different directions started to be explored. For example, learning new relation types from just a few instances (few-shot RE; Han et al., 2018; Gao et al., 2019; Sabo et al., 2021; Popovic and Färber, 2022), or evaluating the models over multiple source domains (cross-domain RE; Bassignana and Plank, 2022b,a). However, a major issue of RE is that most research so far involves the English language only.

After the very first multi-lingual work from the previous decade—the ACE dataset (Doddington et al., 2004) including English, Arabic and Chinese—recent work has started again exploring multi-lingual RE. Seganti et al., 2021 published a multi-lingual dataset, built from entity translations and Wikipedia alignments from the original English version. The latter was collected from automatic alignment between DBpedia and Wikipedia. The result includes 14 languages, but with very diverse relation type distributions: Only English contains instances of all the 36 types, while the most low-resource Ukrainian contains only 7 of them (including the 'no_relation'). This setup makes it hard to directly compare the performance on different languages. Kassner et al., 2021 translated TREx (Elsahar et al., 2018) and GoogleRE,[1] both consisting of triplets in the form (object, relation, subject) with the aim of investigating the knowledge present in pre-trained language models by querying them via fixed templates. In the field of distantly supervised RE, Köksal and Özgür, 2020 and Bhartiya et al., 2022 introduce new datasets including respectively four and three languages in addition to English.

In this paper, we propose MULTI-CROSSRE, to the best of our knowledge the most diverse RE dataset to date, including 27 languages and six diverse text domains for each of them. We automatically translated CrossRE (Bassignana and Plank, 2022a), a fully manually-annotated multi-domain RE corpus, annotated at sentence level. We release the baseline results on the proposed dataset and, as quality check, on the 26 back-translations to English. Additionally, we report an analysis where native speakers in seven diverse languages manually check more than 200 translated sentences and the respective entities, on which the semantic relations are based. MULTI-CROSSRE allows for the investigation of sentence-level RE in the 27 languages included in it, and for direct performance comparison between them. Our contributions are: ① We propose a practical approach to machine-

[1] https://code.google.com/archive/p/relation-extraction-corpus

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 80–85

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with learning algorithms that analyze data used for classification and regression analysis.

Beim maschinellen Lernen sind Support-Vektor-Maschinen (SVMs, auch Support-Vektor-Netzwerke) überwachte Lernmodelle mit Lernalgorithmen, die Daten für Klassifizierungs- und Regressionsanalysen analysieren.

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with learning algorithms that analyse data for classification and regression analysis.

Figure 1: **Example sentence with color-coded entity markup.** From top to bottom: The original English text, its translation to German, and translation back to English. In the first translation step the entity *classification* is not transferred to German. In the second translation step the entity *machine learning* is (wrongly) expanded by a comma—later corrected in our post-processing.

	SENTENCES				RELATIONS			
	train	dev	test	**tot.**	train	dev	test	**tot.**
📰	164	350	400	914	175	300	396	871
🏛	101	350	400	851	502	1,616	1,831	3,949
🍃	103	351	400	854	355	1,340	1,393	3,088
🎵	100	350	399	849	496	1,861	2,333	4,690
📖	100	400	416	916	397	1,539	1,591	3,527
🤖	100	350	431	881	350	1,006	1,127	2,483
tot.	668	2,151	2,446	**5,265**	2,275	7,662	8,671	**18,608**

Table 1: **CrossRE Statistics.** Number of sentences and number of relations for each domain.

translate datasets with span-based annotations and apply it to produce MULTI-CROSSRE, the first multi-lingual and multi-domain dataset for RE including 27 languages and six text domains.[2] ② Multi-lingual and multi-domain baselines over the proposed dataset. ③ Comprehensive experiments over the back-translations to English. ④ A manual analysis by native speakers over more than 200 sentences in seven diverse languages.

2 MULTI-CROSSRE

CrossRE As English base, we use CrossRE (Bassignana and Plank, 2022a),[3] a recently published multi-domain dataset. CrossRE is entirely manually-annotated, and includes 17 relation types spanning over six diverse text domains: artificial intelligence (🤖), literature (📖), music (🎵), news (📰), politics (🏛), natural science (🍃). The dataset was annotated on top of CrossNER (Liu et al., 2021), a Named Entity Recognition (NER) dataset. Table 1 reports the statistics of CrossRE.

Translation Process With the recent progress in the quality of machine translation (MT), utilizing machine-translated datasets in training and evaluation of NLP methods has become a standard practice (Conneau et al., 2018; Kassner et al., 2021). As long as the annotation is not span-bound, producing a machine-translated dataset is rather straightforward. The task however becomes more involved for datasets with annotated spans, such as the named entities in our case of the CrossRE dataset, or e.g. the answer spans in a typical question answering (QA) dataset. Numerous methods have been developed for transferring span information between the source and target texts (Chen et al., 2022). These methods are often tedious and in many cases rely on language-specific resources to obtain the necessary mapping. Some methods also require access to the inner state of the MT system, e.g. its attention activations, which is generally not available when commercial MT systems are used.

In this work, we demonstrate a practical and simple approach to the task of machine translating a span-based dataset. We capitalize on the fact that DeepL,[4] a commercial machine translation service very popular among users thanks to its excellent translation output quality, is capable of translating document markup. This feature is crucial for professional translators—the intended users of the service—who need to translate not only the text of the source documents, but also preserve their formatting. In practice, this means that the input of DeepL can be a textual document with formatting (a Word document) and the service produces its translated version with the formatting preserved.

For the CrossRE dataset, we only need to transfer the named entities, which can be trivially encoded as colored text spans in the input documents, where the color differentiates the individual entities. This is further facilitated by the fact that the entities do not overlap in the dataset, allowing for a simple one-to-one id-color mapping. Observing that oftentimes the entities are over-

[2] https://github.com/mainlp/CrossRE

[3] Released with a GNU General Public License v3.0.

[4] https://www.deepl.com/translator

	lang2vec	TRANSLATION (EN → X)							BACK-TRANSLATION (X → EN) EVAL ON BACK-TRANSLATED DATA							BACK-TRANSLATION (X → EN) EVAL ON ORIGINAL CROSSRE DATA							Δ_{BT}	Δ_{OR}
Language								**avg.**							**avg.**							**avg.**		
German	0.18	24.6	27.6	29.6	9.7	19.7	21.1	22.0	24.9	31.5	27.9	10.5	19.3	21.2	22.5	25.1	30.7	27.7	10.4	19.6	21.5	22.5	0.0	0.8
Danish	0.18	25.5	30.8	33.0	11.9	19.8	21.4	23.7	25.6	31.4	34.6	8.4	20.0	21.4	23.6	25.6	30.6	33.8	8.6	20.1	20.6	23.2	0.4	0.1
Portuguese_BR	0.18	26.2	30.7	29.2	10.7	20.0	21.2	23.0	24.9	34.7	32.1	10.1	18.2	21.5	23.6	25.3	32.5	32.5	10.1	17.9	21.4	23.3	0.3	0.0
Portuguese_PT	0.18	28.2	32.9	31.7	10.5	20.1	22.9	24.4	24.4	34.7	28.0	10.1	19.9	21.9	23.2	25.1	34.5	28.9	10.0	19.7	22.3	23.4	0.2	0.1
Dutch	0.19	25.8	30.9	29.3	9.7	18.5	20.7	22.5	25.0	32.1	30.3	10.5	19.9	21.6	23.2	25.7	32.2	30.3	10.7	20.4	21.8	23.5	0.3	0.2
Ukrainian	0.21	26.7	29.1	27.5	9.0	19.4	20.4	22.0	24.8	31.4	29.9	10.4	16.1	22.5	22.5	24.6	30.9	30.5	10.8	16.2	23.3	22.7	0.2	0.6
Swedish	0.21	25.8	33.4	31.1	10.6	18.6	21.6	23.5	25.7	32.1	33.4	8.0	17.4	20.5	22.9	25.2	31.3	32.4	8.3	17.8	20.2	22.5	0.4	0.8
Slovenian	0.22	27.0	32.3	28.1	7.9	15.0	20.1	21.7	25.3	32.4	28.4	10.5	19.8	21.1	22.9	25.1	31.3	30.2	10.1	20.0	20.2	22.8	0.1	0.5
Italian	0.22	27.1	32.5	31.3	12.8	19.1	22.3	24.2	26.3	34.6	32.0	11.3	19.9	19.7	24.0	26.7	34.3	31.5	11.3	20.2	20.0	24.0	0.0	0.7
Romanian	0.23	26.5	33.0	30.2	10.3	16.6	21.3	23.0	24.0	33.7	29.8	10.8	20.7	19.4	23.1	24.3	30.5	30.4	10.8	20.0	19.2	22.5	0.6	0.8
Bulgarian	0.23	28.1	34.4	27.2	9.0	20.4	20.9	23.3	24.3	31.5	29.2	10.8	19.1	21.4	22.7	24.3	31.1	30.9	10.9	19.0	21.5	22.9	0.2	0.4
French	0.23	29.6	33.5	32.3	11.3	19.3	23.5	24.9	25.5	33.5	31.4	11.2	19.8	21.8	23.9	25.5	32.1	31.2	10.9	20.1	21.7	23.6	0.3	0.3
Slovak	0.23	23.1	32.7	28.2	9.2	18.6	18.2	21.7	24.4	32.6	31.6	10.2	19.2	19.8	23.0	24.1	33.6	31.7	10.3	17.8	20.1	22.9	0.1	0.4
Indonesian	0.24	26.0	34.6	33.2	9.6	19.7	20.7	24.0	25.2	32.9	32.6	9.7	16.9	20.9	23.0	26.1	32.9	32.4	9.8	16.5	20.7	23.1	0.1	0.2
Latvian	0.25	24.8	32.3	25.0	11.0	15.9	19.1	21.4	24.3	32.6	27.6	8.7	18.8	20.5	22.1	24.4	30.9	28.7	8.5	19.1	20.5	22.0	0.1	1.3
Spanish	0.27	27.6	32.2	29.9	9.7	19.2	22.5	23.5	24.5	32.4	29.1	9.2	19.5	23.9	23.1	24.6	31.9	28.6	9.5	20.2	23.3	23.0	0.1	0.3
Hungarian	0.27	22.4	28.9	26.0	8.5	19.2	18.4	20.6	21.2	31.0	28.5	8.6	18.5	21.2	21.5	22.2	30.2	29.1	8.5	19.3	21.3	21.8	0.3	1.5
Greek	0.27	28.3	33.3	31.8	9.1	20.3	22.7	24.2	24.1	30.7	32.9	11.2	18.6	19.8	22.9	24.7	31.9	33.6	10.9	19.2	20.8	23.5	0.6	0.2
Estonian	0.27	23.4	29.3	27.4	8.3	17.1	19.0	20.8	22.7	31.8	29.2	8.5	15.8	19.4	21.2	23.8	30.6	30.4	8.5	16.4	18.4	21.3	0.1	2.0
Lithuanian	0.27	26.2	31.5	26.3	9.9	18.9	16.2	21.5	24.5	31.3	26.4	10.8	18.8	21.4	22.2	25.3	30.0	27.6	10.3	18.6	21.2	22.2	0.0	1.1
Polish	0.27	24.6	34.3	28.7	10.4	19.5	19.9	22.9	24.4	31.6	27.9	9.7	16.6	20.4	21.8	24.5	30.9	28.6	9.6	16.6	20.8	21.8	0.0	1.5
Finnish	0.28	22.9	30.2	24.7	8.8	17.0	18.1	20.3	21.4	29.5	27.1	8.8	17.4	20.5	20.8	24.9	34.7	32.1	10.1	18.2	21.5	23.6	2.8	0.3
Czech	0.29	25.0	30.1	28.4	10.1	19.4	18.1	21.8	23.8	30.8	29.0	9.8	20.2	19.6	22.2	24.4	31.9	29.5	9.7	19.6	20.0	22.5	0.3	0.8
Chinese	0.30	22.2	33.4	25.0	9.0	20.1	18.7	21.4	23.1	28.4	27.1	9.5	18.9	22.0	21.5	23.8	28.7	27.4	9.9	18.7	21.3	21.6	0.1	1.7
Turkish	0.38	23.8	29.4	26.7	10.6	20.4	18.2	21.5	23.4	23.2	28.4	9.3	17.6	19.1	20.2	24.5	23.2	29.8	9.2	17.9	20.3	20.8	0.6	2.5
Japanese	0.41	22.6	29.2	20.1	8.9	19.5	12.9	18.9	21.1	27.4	21.7	8.0	16.1	15.2	18.3	20.5	27.9	23.4	8.1	16.1	16.2	18.7	0.4	4.6

Table 2: **MULTI-CROSSRE Baseline Results.** Macro-F1 scores of the baseline model ordered by increasing lang2vec distance from English. Δ_{BT}: delta between back-translated and original evaluation when model trained on back-translated data. Δ_{OR}: delta between model trained on back-translated data and on original CrossRE data when evaluated on original CrossRE English.

							avg.
English	20.8	36.4	30.7	10.1	20.0	21.6	23.3

Table 3: **CrossRE Baseline Results.** Macro-F1 scores of the RC baseline over the original CrossRE English dataset.

extended by a punctuation symbol during translation, the only post-processing we apply is to strip from each translated entity any trailing punctuation not encountered in the suffix of the original named entity. The process is illustrated in Figure 1, with details about two typical issues with this approach (later analysed in Section 4).[5]

3 Experiments

Model Setup In order to be able to directly compare our results with the original CrossRE baselines on English, we follow the model and task setup used by Bassignana and Plank, 2022a. We perform Relation Classification (Han et al., 2018; Baldini Soares et al., 2019; Gao et al., 2019), which consists of assigning the correct relation types to the ordered entity pairs which are given as semantically connected. The model follows the current state-of-the-art architecture by Baldini Soares et al., 2019 which augments the sentence with four entity markers e_1^{start}, e_1^{end}, e_2^{start}, e_2^{end} surrounding the two entities. Following Zhong and Chen (2021) the entity markers are enriched with information about the entity types. The augmented sentence is then passed through a pre-trained encoder (XLM-R large; Conneau et al., 2020), and the classification made by a linear layer over the concatenation of the start markers $[\hat{s}_{e_1^{start}}, \hat{s}_{e_2^{start}}]$. We run all our experiments over five random seeds. See Appendix A for reproducibility and hyperparameters settings.

Results The original CrossRE study reports the baseline experiments by using the mono-lingual BERT (Devlin et al., 2019) language encoder. In order to be able to compare the original baseline with the results on our MULTI-CROSSRE dataset, we re-run the English experiments by using the multi-lingual XLM-R large (Conneau et al., 2020) language encoder, and report the results in Table 3.

In Table 2 we report the results of our experiments over MULTI-CROSSRE. The left-most columns are the results of the models trained and evaluated over the translated data (from English to language X). As a sanity check, we back-translated the data from each of the 26 new languages to English (from language X to English). We train and evaluate new models on this data in the middle columns. Finally, on the right-most

[5]The overall translation process cost is ≈ 60€.

columns we evaluate the same models—trained on back-translated data—over the original CrossRE test sets. We sort the languages by increasing distance to English, computed as the cosine distance between the syntax, phonology and inventory vectors of lang2vec (Littell et al., 2017).

For our analysis we consider the average of the six domains.[6] Our scores on the translated data reveal a relatively small drop in respect to the English baseline in Table 3. The difference range goes from an improvement of $+1.6$ Macro-F1 points on French, to a maximum drop of -4.4 on Japanese—which has the largest lang2vec distance with respect to English (0.41). The results of the models trained on the back-translated data present essentially the same trend between evaluating on the back-translations and on the original CrossRE English data—with a Pearson's correlation coefficient of 0.88—confirming the high quality of the proposed translation. The only exception if Finnish, with a difference of 2.8 points between the two evaluations. All the other languages report a smaller difference in a range between 0.0 and 0.6. The lang2vec distance is not informative of the quality of the individual translations (Pearson's correlation -0.59). However, other factors should be taken into account, e.g. the language model performances on each individual language.

4 Manual Analysis

We performed a manual analysis for further inspecting the quality and usability of MULTI-CROSSRE for studying multi-lingual RE. We manually checked 210 sentences from a diverse set of seven languages, including one North Germanic (Danish), one Uralic (Finnish), one West Slavic (Czech), two Germanic (German and Dutch), one Latin (Italian), and one Japonic (Japanese). For each of them, native speakers annotated the following: ① In how many sentences is the overall meaning preserved? ② How many entities are transferred to language X? ③ How many entities are correctly translated? ④ How many entities are marked with the correct entity boundaries?

We annotated 30 sentences for each language. Table 4 reports the statistics of our analysis. Overall, we find a surprisingly high quality of entity translations (96% are judged as correct by our human annotators). Out of the seven languages, Japanese is the one suffering the most by the translation process and, as we discussed above, this is reflected in the lowest scores in Table 2. Some entities are not transferred. These are mostly due to compounds typical for some languages. For example, the English snippet "the *Nobel* laureate" (where only *Nobel* is marked as entity), is translated to Danish as "nobelpristageren", and to Dutch as "Nobelprijswinnaar". In Italian, which in this regard behaves more similarly to English, all the entities are correctly transferred. In Appendix B we report the total per-language percentages of transferred entities and relations. Regarding the entity translations and the entity boundaries, the latter is a bigger challenge for the translation tool, often including surrounding function words—e.g. the writer *Pat Barker* in Danish is extended to the entity *Pat Barker er*. These could easily be post-processed, but since the Relation Classification model relies on the injected entity markers, it is not much influenced by this type of error (see baseline discussion in Section 3).

Language	Sent. Transl.	# entities	Ent. Transl.	Ent. Bound.
English	30	160	-	-
Czech	28	158	152	143
Danish	27	158	143	136
Dutch	28	158	156	141
Finnish	30	150	141	137
German	27	151	148	139
Italian	29	160	157	152
Japanese	19	150	145	82

Table 4: **Statistics of the Manual Analysis.** At the top, total amount of original English sentences and annotated entities within them. Below, for each sample set, amount of correct instances in the four categories of sentence translation, number of entities, entity translations, and entity boundaries.

5 Conclusion

We introduce MULTI-CROSSRE, the most diverse RE dataset to date, including 26 languages in addition to the original English, and six text domains. The proposed span-based MT approach could be easily applied to similar cases. We report baseline results on the proposed resource and, as quality check, we back-translate MULTI-CROSSRE to English and run the baseline model again over it. Our manual analysis reveals that the higher challenge during the translation is transferring the correct entity boundaries. However, given the model architecture, this does not influence the scores.

[6] Bassignana and Plank, 2022a discuss the lower scores of news (🗞) attributing them to the data coming from a different data source and the fewer amount of relation instances with respect to the other domains.

Acknowledgments

We thank the MaiNLP/NLPnorth group for feedback on an earlier version of this paper, and ITU's High-performance Computing cluster for computing resources.

EB and BP are supported by the Independent Research Fund Denmark (Danmarks Frie Forskningsfond; DFF) Sapere Aude grant 9063-00077B. BP is supported by the ERC Consolidator Grant DIALECT 101043235. FG and SP were supported by the Academy of Finland.

References

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Elisa Bassignana and Barbara Plank. 2022a. CrossRE: A cross-domain dataset for relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3592–3604, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Elisa Bassignana and Barbara Plank. 2022b. What do you mean by relation extraction? a survey on datasets and study on scientific relation classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 67–83, Dublin, Ireland. Association for Computational Linguistics.

Abhyuday Bhartiya, Kartikeya Badola, and Mausam . 2022. DiS-ReX: A multilingual dataset for distantly supervised relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 849–863, Dublin, Ireland. Association for Computational Linguistics.

Yang Chen, Chao Jiang, Alan Ritter, and Wei Xu. 2022. Frustratingly easy label projection for cross-lingual transfer.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6250–6255, Hong Kong, China. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Nora Kassner, Philipp Dufter, and Hinrich Schütze. 2021. Multilingual LAMA: Investigating knowledge in multilingual pretrained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3250–3258, Online. Association for Computational Linguistics.

Abdullatif Köksal and Arzucan Özgür. 2020. The RELX dataset and matching the multilingual blanks for cross-lingual relation classification. In *Findings*

of the Association for Computational Linguistics: EMNLP 2020, pages 340–350, Online. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021. Crossner: Evaluating cross-domain named entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13452–13460.

Nicholas Popovic and Michael Färber. 2022. Few-shot document-level relation extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5733–5746, Seattle, United States. Association for Computational Linguistics.

Ofer Sabo, Yanai Elazar, Yoav Goldberg, and Ido Dagan. 2021. Revisiting few-shot relation classification: Evaluation data and classification schemes. *Transactions of the Association for Computational Linguistics*, 9:691–706.

Alessandro Seganti, Klaudia Firlag, Helena Skowronska, Michał Satława, and Piotr Andruszkiewicz. 2021. Multilingual entity and relation extraction dataset and model. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1946–1955, Online. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

Appendix

A Reproducibility

We report in Table 5 the hyperparameter setting of our RC model (see Section 3). All experiments were ran on an NVIDIA® A100 SXM4 40 GB GPU and an AMD EPYC™ 7662 64-Core CPU.

Parameter	Value
Encoder	`xlm-roberta-large`
Classifier	1-layer FFNN
Loss	Cross Entropy
Optimizer	Adam optimizer
Learning rate	$2e^{-5}$
Batch size	32
Seeds	4012, 5096, 8878, 8857, 9908

Table 5: **Hyperparameters Setting.** Model details for reproducibility of the baseline.

Language	% Entities	% Relations
German	96.7	91.4
Danish	97.5	93.9
Portuguese_BR	99.8	99.5
Portuguese_PT	99.8	99.6
Dutch	98.5	95.8
Ukrainian	99.1	97.7
Swedish	97.6	94.1
Slovenian	99.1	98.0
Italian	99.8	99.5
Romanian	98.8	96.7
Bulgarian	99.5	98.9
French	99.6	99.4
Slovak	99.2	98.1
Indonesian	99.8	99.5
Latvian	99.4	98.6
Spanish	99.3	98.3
Hungarian	98.2	95.8
Greek	98.8	98.0
Estonian	97.9	94.6
Lithuanian	99.4	98.8
Polish	99.4	98.6
Finnish	96.0	90.7
Czech	99.0	98.0
Chinese	99.3	98.4
Turkish	99.4	98.5
Japanese	94.9	88.9

Table 6: **Transferred Entities and Relations.** Percentages of entities and of relations transferred during the translation process for each language.

B Per-language Analysis

In table 6 we report the percentages of entities which are transferred during the translation process from the original English to language X, and the percentage of relations which do not involve missing entities (i.e. are transferred during the translation process).

Microservices at Your Service: Bridging the Gap between NLP Research and Industry

Tiina Lindh-Knuutila[1], **Hrafn Loftsson**[2], **Pedro Alonso Doval**[3], **Sebastian Andersson**[1], **Bjarni Barkarson**[2], **Héctor Cerezo-Costas**[3], **Jón Guðnason**[2], **Jökull Snær Gylfason**[2] **Jarmo Hemminki**[1], **Heiki-Jaan Kaalep**[4]

[1] Lingsoft, Turku, Finland
[2] Reykjavik University, Reykjavik, Iceland
[3] Gradiant, Pontevedra, Spain
[4] University of Tartu, Tartu, Estonia

{tiina.lindh-knuutila, sebastian.andersson, jarmo.hemminki}@lingsoft.fi
{hrafn, bjarnibar, jg, jokullg}@ru.is
{palonso, hcerezo}@gradient.org
{heiki-jaan.kaalep}@ut.ee

Abstract

This paper describes a collaborative European project whose aim was to gather open source Natural Language Processing (NLP) tools and make them accessible as running services and easy to try out in the European Language Grid (ELG). The motivation of the project was to increase accessibility for more European languages and make it easier for developers to use the underlying tools in their own applications. The project resulted in the containerization of 60 existing NLP tools for 16 languages, all of which are now currently running as easily testable services in the ELG platform.

1 Introduction

Universities and other research institutes in Europe, and sometimes companies, are nowadays often publishing open source Natural Language Processing (NLP) software on various platforms, primarily GitHub. This software is often associated with research papers and, in the best case, also linked to other sharing platforms, such as CLARIN[1] or META-SHARE[2]. GitHub is, however, often the only place in which the tools are available. If a user finds a tool with a suitable license, it may still be difficult to determine if the tool works as intended. The threshold for trying out these NLP tools can also be high due to the reliance on various dependencies that may not be compatible with other desired tools or the tools are simply not up to date. Reproducibility of results is important in NLP but currently many results cannot be reproduced, even if the code is available. For example, Wieling et al. (2018) were only able to reproduce the same results in 1 out of 10 experiments.

In this paper, we describe a collaborative European project, *Microservices at Your Service: Bridging the Gap between NLP Research and Industry*[3] (hereafter simply referred to as the *Microservices* project), carried out by four partners: Lingsoft, a private company from Finland, University of Tartu from Estonia, Reykjavik University from Iceland, and Gradiant, a non-profit organisation from Spain. The main aim of the project was to increase accessibility of NLP tools for more European languages by:

- Making the tools available as running services in the European Language Grid[4] (ELG), and, additionally, registering them in ELRC-SHARE[5] for higher visibility and reach.
- Providing, for each tool, a tested container image which takes care of any dependencies and provides a logical handling of the data inputs and outputs, should the users want to use the container in their own computing environment.

[1] http://clarin.eu/
[2] http://www.meta-share.org/
[3] https://www.lingsoft.fi/en/microservices-at-your-service-bridging-gap-between-nlp-research-and-industry
[4] https://live.european-language-grid.eu/
[5] https://elrc-share.eu/

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 86–91
May 22-24, 2023

- Providing training and dissemination in the form of recorded workshops about the containerization of the tools, uploading the tools to the ELG.

- Finally, showcasing how the tools can be integrated for different purposes.

The project has deployed services in the ELG for 16 languages (see Section 4). For many languages, there is a distinct lack of resources in the current academic NLP research (Maria Giagkou, 2022). Highlighting the efforts made for low-resource languages is paramount to foster the development and usage of these resources by both the academic community and the industry, and our project targeted several of these low-resource languages.

Each underlying open source tool was implemented as a microservice (see Section 2.1) using Docker (see Section 3.3) for containerization. This allows developers, who need functionality from the various tools, to design their NLP applications as a collection of loosely coupled running services, as opposed to building the application using sources from various Github repositories, which, notably, may be written using various programming languages and depend on various external libraries.

In total, our project has resulted in the containerization of 60 existing NLP tools, all of which are currently running as services accessible through the ELG.

2 Background

Nowadays, software is often distributed to the end users via the Internet, rather than having the users install the software on their local machines. This method of distribution is called software-as-a-service or SaaS. Many large commercial organisations offer cloud platforms for distributing software, e.g. AI and NLP as SaaS, to the end users, and on some platforms it is possible for other organisations than the platform provider to upload their own tools for further distribution.

In this section, we provide the reader with basic information on the concept of microservices, the ELG cloud platform, and ELRC-SHARE.

2.1 Microservices

The microservice architectural style for software development has been defined as "[..] an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API" (Lewis and Fowler, 2014).

One of the advantages of microservices is that they can be updated without the need of redeploying the application that uses them. Another advantage is that different services can be implemented in different programming languages. In the contrasting monolithic architectural style, an application is built as a single executable unit (often using a single programming language). Any changes to the functionality demand building and deploying a new version of the application.

According to Francesco et al. (2017), "[m]icroservice architectures are particularly suitable for cloud infrastructures, as they greatly benefit from the elasticity and rapid provisioning of resources."

2.2 European Language Grid

The ELG is a scalable cloud platform, which hosts tools, data sets, and records of Language Technology (LT) projects and LT providers in official 24 EU languages and many additional ones. The goal of the ELG is to become the primary platform for LT, including NLP and speech technologies, in Europe. An important part of the purpose of ELG is to support digital language equality, "i.e., to create a situation in which all languages are supported through technologies equally well" (Rehm et al., 2021). Additionally, there is a growing movement to ensure that all relevant services can be offered by European providers to improve EU-wide digital sovereignty (European Parliament et al., 2023). Currently, most European cloud services are provided by non-European providers (Synergy Research Group, 2022).

The ELG platform is growing continuously and they foresee a need to evolve in the following areas: hardware capacity and cost distribution, hardware acceleration (for example, there is no GPU support yet), integration and deployment support, and workflow support (Kintzel et al., 2023).

ELG provides resources for developers to easily integrate a service: A (micro)service running in the ELG is wrapped with the ELG LT Service API and packaged in a Docker container. Both of these steps are carried out by the developer of the service. Thereafter, the container is integrated into

the ELG: It can either be called through the API or tested using a web UI. All APIs are https-based and use JSON as the primary data representation format. For easy creation of an application for an ELG-compatible service, Java- and Python-based libraries are available (Galanis et al., 2023).

For a user looking for potential tools, the ELG platform provides a faceted search functionality, allowing search by resource type such as corpus, tool, functionality, availability as an ELG-compatible service, data type, language, and license in a simple manner. The submissions to the service are also validated, which should improve the findability compared to a platform without such validation process.

2.3 ELRC-SHARE

ELRC-SHARE is a repository, maintained by the European Language Resource Coordination (ELRC)[6], for documenting, storing and accessing language data and tools in all EU languages, Norwegian Bokmål, Norwegian Nynorsk, and Icelandic. The original intent of the repository was to obtain and store data and tools that contribute to the European Commission's automated `eTranslation` platform[7], but the scope has broadened to include other LT tools as well. Approximately 80% of the language resources are freely usable outside ELRC (Marra et al., 2022).

3 Project Execution

Our two year project started in March 2021. The goal of the project (described in Section 1) included several stages. In the first stage, we sought out open source tools that might be of potential interest. We prioritized those that are actively maintained or developed. This was carried out both by bottom-up search on the software sharing platforms (primarily GitHub), and by contacting research institutions in the targeted regions. In parallel, we also collected standard or available test data sets for the tools. This initial phase was followed by testing the set of collected tools on the existing test data. If many tools existed for the same task, a selection was made based on metrics performance and language coverage. After all tools were tested and selected, we started containerizing the tools and expose a web service API for each of them on the ELG. Finally, we stored metadata information of each tool in ELRC-SHARE. Our dissemination activities ran parallel to making the tools available: We held workshops on different themes of the project, ranging from dockerization of the tools to demonstrating their functionality and use case integration.

3.1 Searching for tools

The search for tools was not primarily guided by pre-specified project goals or use cases, but rather guided by the subjective explorative interests of the individual partners.

At the start of the project, there was an initial assumption made that university labs or individual programmers were storing interesting and useful tools on local disks. These tools could then be made public via the project. However, the reality was different: source code was always in GitHub[8] or GitLab[9]. The focus therefore quickly shifted to verifying that the found tools were functioning well.

To find interesting tools, we sent emails to university contacts, browsed university web repositories and arXiv, did online searches with relevant keywords (e.g. 'speech recognition', 'parsing', or 'named entity recognition') and looked up conference proceedings and journal articles for interesting repositories. Then, we went through each promising repository to see first if all the relevant parts for running the tool were available. This was followed by an initial compilation of the tool and ensuring that we obtained the same or at least similar results as the original authors, if the test data was available. If not, we gathered examples to ensure the test results seemed reasonable.

3.2 Testing and documenting

To make a third-party tool available for the wider public involves providing documentation, which minimally describes the following: a) What the purpose of the tool is; b) how to run the tool; c) specification of the tool input and output formats and error handling; d) the original authors of the tool; and e) what kind of a licence or terms of use the tool has.

Often these points have already been addressed by the authors of the tool, although the amount of details varied. We sometimes had to fill in missing

[6] `https://www.lr-coordination.eu/`

[7] `https://webgate.ec.europa.eu/etranslation/public/welcome.html`

[8] `http://github.com/`

[9] `http://gitlab.com/`

information (most notably the licence) and come up with our own wording about the purpose and place of the tool in the ecosystem of the LT field of the particular language.

While creating the documentation for the microservices tools it was noticed that some tools with similar functionality had differing output types without an explicit reason why. Such differences can of course be justified, but can also indicate that some standardisation in a field might benefit interoperability. This was especially notable for morpho-syntactic categories for Estonian and University of Tartu set up a designated webpage[10] for facilitating comparison between these identified systems.

3.3 Dockerization

We used Docker[11] for developing, distributing and running the NLP tools (in the ELG). Docker has in recent years been established as a convenient solution for making it easier to create, deploy, and run applications by using containers. Containers allow developers to package up an application with all requirements, such as libraries and other dependencies, and distribute it as a single stand-alone package. Docker is a good option for a platform independent solution for making NLP tools available for both researchers and software developers.

Each of the selected NLP tools was dockerized by building a container with the tool itself along with an http API that gives people/programs outside the container access to the tool. All of the images for our tools are shared in the Docker Hub[12], world's largest library for container images. The difficulty of dockerizing a given NLP tools was dependent on how easy it was to give the API in the container access to the tool. Once the API was able to receive output from the NLP tool, all that was left was to make sure that the output from the API was in accordance to the ELG specification.

For each service integrated to the ELG, we also provided metadata, which contains a link to the code repository of the underlying tool.

4 The NLP Tools

In our project, the focus was on tools for the Nordic/Scandinavian languages, the Baltic languages, and the Iberian languages, simply because of the partners' geographical locations and local interests.

We dockerized 60 existing NLP tools, in 16 languages: Catalan: 2; English: 2; Estonian: 11; Faroese: 1; Finnish: 4; Galician: 1; Basque: 1; Icelandic: 11; Komi: 1; Latvian: 3; Lithuanian: 2; Northern Sami: 2; Norwegian: 1; Portuguese: 6; Spanish: 5; and Swedish: 3. Additionally, we provided four multilingual tools. Whilst the majority these tools come from European institutions, the project also made available relevant results from South American countries (Brazil, Chile and Uruguay).

The list of dockerized tools is available at the project website. The NLP tools are very diverse, covering from low level (e.g. PoS taggers, morphological analyzers, NERs and parsers) to high level applications (e.g. question answering (QA) and audio processing), as well as others with niche results (detection of false friends and text generation of proverbs given a short text)

5 Getting the Tools into Use

There is a risk that new tools made for low-resource languages might not be known by the community. A tool might be created as a one time release for an academic publication, or it might not have gathered the attention needed for a continued development. For the purpose of both stimulating researchers to share their tools and promote the tools we made available, we held three types of workshops: First we had an early awareness workshop, in which we provided hands-on guidance on how to release available tools as Docker images. During the second year, we held two workshops focusing on how to make tools available in the ELG platform. Finally, at the end of the project, we held workshops which summarized our work and demonstrated how the tools we provided can be integrated into LT applications. All workshops are made available on the project webpage.

In what follows, we describe some of these pilot integration cases. In each of these cases, it was easy to "plug in" a container with a well defined API, and then handle the input and output in the process pipeline.

A language identification (LID) tool was utilized in two different cases. In a translation process, we utilized it to make sure the training data for a neural machine translation (NMT) model was actually in the correct language. The original

[10] https://cl.ut.ee/ressursid/morfo-systeemid/

[11] https://www.docker.com/

[12] https://hub.docker.com/

texts contained sentences in other languages, causing an in-production NMT model to occasionally produce English instead of Swedish translations. The previous LID tool had a 98.9 % precision and 96.4% recall for Swedish, whereas the new tool, HeLI OTS (Jauhiainen et al., 2022), had a 99.9 % precision and 99.6% recall. When there are hundreds of millions of words in the training material, one percentage point yields millions of words tagged in wrong language. The new LID tool alleviated this problem to a sufficient extent.

This LID tool was also found useful in an online library platform[13], where publishers provide large amounts of e-books. Sometimes the metadata provided by the publisher does not match the language of the actual e-book, yielding erroneous behavior, for example, in screen readers.

At the online library platform, we also piloted aligning audio books and e-books, to allow seamless switching from text to audio and back, using an audio alignment tool. This tool was not designed for this kind of task originally, but, nevertheless, it allowed testing potential new features for the platform. Furthermore, we also tested NER and linking to ontologies to further improve the findability within an e-book or audio book.

6 Limitations

In the previous sections, we have argued that it can be beneficial to dockerize NLP tools for the purpose of making them accessible as running microservices. However, this approach can have some practical limitations.

First, changes to a tool do not automatically become available in the dockerized version. Thus, the running microservice in the ELG might become outdated. However, if the developer of the underlying tool is keen on making the newest version running as a microservice, the developer can easily build the docker image again (the code for building it is open source) and then ask ELG to pull the new image from the associated docker hub. Most of that process can also be automated.

Second, due to resource constraints, ELG services are not guaranteed to be constantly running. If a user calls the API of a service, which is not running, the user will probably experience considerable initial delay (associated with the first API call) before the requested service has started.

[13] https://www.ellibs.com

With regard to both of the above mentioned limitations, it is worth noting that anyone can use a given docker image to expose an API for the underlying tool on some web server. In other words, ELG is not the only option for providing access to a running service.

7 Conclusion

In this paper, we have described a collaborative project which succeeded in making 60 NLP tools covering a total of 16 languages available as microservices in the ELG platform. We also described the microservice principles and the European platforms that record or host these microservices, and the steps to get the tools into these platforms.

We recommend that researchers continue this work by providing their tools as Docker images and as compatible services in the ELG platform. This requires just a little more effort from the researchers, but substantially lowers the threshold for testing the tool for new researchers/developers. Hence, lowering the threshold for integrating the tool in new services and raising the potential impact of the initial research.

Acknowledgments

This project has been co-financed by the Connecting Europe Facility of the European Union, project number 2020-EU-IA-0046.

References

Paolo Di Francesco, Ivano Malavolta, and Patricia Lago. 2017. Research on Architecting Microservices: Trends, Focus, and Potential for Industrial Adoption. In *2017 IEEE International Conference on Software Architecture (ICSA)*, pages 21–30.

Dimitris Galanis, Penny Labropoulou, Ian Roberts, Miltos Deligiannis, Leon Voukoutis, Katerina Gkirtzou, Rémi Calizzano, Athanasia Kolovou, Dimitris Gkoumas, and Stelios Piperidis. 2023. Contributing to the European Language Grid as a Provider. In *European Language Grid: A Language Technology Platform for Multilingual Europe*, pages 67–93, Online. Springer International Publishing.

Synergy Research Group. 2022. European Cloud Providers Continue to Grow but Still Lose Market Share. Accessed: 05-04-2023.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2022. HeLI-OTS, Off-the-shelf Language Identifier for Text. In *Proceedings of the 13th Conference on Language Resources and Evaluation*,

pages 3912–3922, Marseille, France. European Language Resources Association.

Florian Kintzel, Rémi Calizzano, and Georg Rehm. 2023. Cloud Infrastructure of the European Language Grid. In *European Language Grid: A Language Technology Platform for Multilingual Europe*, pages 95–106, Online. Springer International Publishing.

James Lewis and Martin Fowler. 2014. Microservices: a definition of this new architectural term. Accessed: 05-04-2023.

Maria Giagkou. 2022. Digital Language Equality in Europe: How are our languages doing? Presentation in Panel for the Future of Science and Technology (STOA), European Union. Accessed: 17-04-2023.

Eileen Marra, Andrea Lösch, Stefania Racioppa, Hélène Mazo, Maria Giagkou, Dimitra Anastasiou, Natassa Avraamides, Carl Frederik Bach Kirchmeier, Yngvil Beyer, António Branco, Virginijus Dadurkevičius, Hristina Dobreva, Rickard Domeij, Jane Dunne, Kristine Eide, Maria Gavriilidou, Stanislava Graf, Dagmar Gromann, Thibault Grouas, Normunds Grūzītis, Jan Hajič, Barbara Heinisch, Veronique Hoste, Simon Krek, Gauti Kristmannsson, Svetla Koeva, Anna Kotarska, Kaisamari Kuhmonen, Krister Lindén, Teresa Lynn, Kinga Matyus, Maite Melero, Laura Mihăilescu, Željka Motika, Tríona Ní Mhathuna, Micheál Ó Conaire, Maciej Ogrodniczuk, Jon Arild Olsen, Michael Rosner, Elisa Schnell, Maria Skeppstedt, Alexandra Soska, Donatienne Spiteri, Marko Tadić, Carole Tiberius, Dan Tufiș, Andrius Utka, Tamás Váradi, Kadri Vare, Andreas Witt, Josephine Worm Andersson, François Yvon, Jānis Ziediņš, Bódi Zoltán, and Miroslav Zumrík. 2022. AI for Multilingual Europe – Why Language Data Matters. White paper, ELRC Consortium. 3rd edition.

European Parliament, Council of the European Union, European Commission, Directorate-General for Communications Networks Content, and Technology. 2023. European Declaration on Digital Rights and Principles for the Digital Decade. *Official Journal of the European Union*, C 23(1). PUB/2023/89.

Georg Rehm, Stelios Piperidis, Kalina Bontcheva, Jan Hajic, Victoria Arranz, Andrejs Vasiljevs, Gerhard Backfried, Jose Manuel Gomez-Perez, Ulrich Germann, Rémi Calizzano, Nils Feldhus, Stefanie Hegele, Florian Kintzel, Katrin Marheinecke, Julian Moreno-Schneider, Dimitris Galanis, Penny Labropoulou, Miltos Deligiannis, Katerina Gkirtzou, Athanasia Kolovou, Dimitris Gkoumas, Leon Voukoutis, Ian Roberts, Jana Hamrlova, Dusan Varis, Lukas Kacena, Khalid Choukri, Valérie Mapelli, Mickaël Rigault, Julija Melnika, Miro Janosik, Katja Prinz, Andres Garcia-Silva, Cristian Berrio, Ondrej Klejch, and Steve Renals. 2021. European Language Grid: A Joint Platform for the European Language Technology Community. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 221–230, Online. Association for Computational Linguistics.

Martijn Wieling, Josine Rawee, and Gertjan van Noord. 2018. Reproducibility in Computational Linguistics: Are We Willing to Share? *Computational Linguistics*, 44(4):641–649.

Slaapte or *Sliep*? Extending Neural-Network Simulations of English Past Tense Learning to Dutch and German

Xiulin Yang **Jingyan Chen** **Arjan van Eerden**

Ahnaf Mozib Samin **Arianna Bisazza**

{x.yang.31, j.chen.63, a.j.van.eerden, a.m.samin}@student.rug.nl
a.bisazza@rug.nl
University of Groningen, The Netherlands

Abstract

This work studies the plausibility of sequence-to-sequence neural networks as models of morphological acquisition by humans. We replicate the findings of Kirov and Cotterell (2018) on the well-known challenge of the English past tense and examine their generalizability to two related but morphologically richer languages, namely Dutch and German. Using a new dataset of English/Dutch/German (ir)regular verb forms, we show that the major findings of Kirov and Cotterell (2018) hold for all three languages, including the observation of over-regularization errors and micro U-shape learning trajectories. At the same time, we observe troublesome cases of non human-like errors similar to those reported by recent follow-up studies with different languages or neural architectures. Finally, we study the possibility of switching to orthographic input in the absence of pronunciation information and show this can have a non-negligible impact on the simulation results, with possibly misleading findings.

1 Introduction

The plausibility of neural network-based or connectionist models in simulating psycholinguistic behaviours has been attracting considerable attention since Rumelhart and McClelland (1986) first modeled the past-tense acquisition with an early example of sequence-to-sequence network. Their experiment received harsh criticism (e.g., Pinker and Prince, 1988) but also inspired cognitive scientists with alternatives (e.g., Kirov and Cotterell, 2018; Plunkett and Juola, 1999; Taatgen and Anderson, 2002). Much more recently, Kirov and Cotterell (2018) replicated Rumelhart and McClelland (1986)'s simulations using a modern encoder-decoder neural architecture developed for the task of morphological paradigm completion. Their improved results resolved much of the original criticisms by Pinker and Prince (1988).

The main purpose of this paper is to study the generalizability of Kirov and Cotterell (2018)'s findings beyond the case of English. Specifically, we consider two languages that are genetically related to English, but morphologically richer – namely, Dutch and German. In these languages too, past tense inflection is divided into regular and irregular verbs, but with different proportions and different inflectional patterns than English. Moreover, German and Dutch are characterized by a much more transparent orthography than English (Van den Bosch et al., 1994; Marjou, 2021), which allows us to study the usability of grapheme-based input for simulating past tense acquisition patterns when pronunciation information may not available. Concretely, we aim to answer the following research questions:

1. Can the model applied by Kirov and Cotterell (2018) to English also simulate the past tense acquisition process in languages with more complex morphological inflection, such as Dutch and German?

2. Given the more predictable grapheme-to-phoneme correspondence, i.e., orthographic transparency (Marjou, 2021), in these two languages, will the model perform similarly if the written forms of verbs are used for training instead of the phonetic ones?

To answer these two questions, we build and release a new past-tense inflection dataset of English, Dutch, and German, covering both grapheme and phoneme features (Section 3).[1] We

[1] All code and data are available at https://github.com/JingyanChen22/IK-NLP-Project-4.git

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 92–102

then replicate the single-task learning experiments of Kirov and Cotterell (2018) (Section 4) and extend them to our multilingual dataset, using both phoneme- and grapheme-based input for comparison (Section 5).

Our findings reconfirm the potential and limitations of using neural networks for the simulation of human language learning patterns. Our model shows human-like behavior in learning past tenses of verbs, such as the micro U-shape coined by Plunkett et al. (1991) and over-regularization errors in all the examined languages; however non human-like errors are also reported. We also find that learning irregular past tense forms is considerably easier in Dutch and German than in English. Finally, we observe that higher orthographic transparency indeed leads to more consistent learning results when a model is trained with grapheme vs. phoneme input.

2 Background

Past tense debate The acquisition of verbal past tense in English, particularly the over-regularization of the irregular verbs in the process of learning (Marcus et al., 1992), has been serving as a testing ground for different hypotheses in language modelling for decades. A much debated question is whether the past tense of (ir)regular verbs is learnt by rules and memories (e.g., Plaut and Gonnerman, 2000; Seidenberg and Gonnerman, 2000; Marcus et al., 1995; Albright and Hayes, 2003; Pinker and Ullman, 2002), by analogy (e.g., Ramscar, 2002; Albright and Hayes, 2003) or by a dual mechanism (Pinker and Prince, 1988; Taatgen and Anderson, 2002).

Marcus et al. (1995) posited the necessity of mental rules in learning German irregular verbs. By contrast, Ernestus and Baayen's (2004) and Hahn and Nakisa's (2000) studies on Dutch and German respectively provided evidence in favour of connectionist and analogical approaches: they showed that humans tend to choose wrong past tense suffixes for regular verbs whose phonological structure is similar to that of irregular ones.

Recent connectionist *revival* The recent development of deep learning methods in computational linguistics has led to a renewed interest in connectionist approaches to modelling language acquisition and processing by humans (e.g., Blything et al., 2018; Kádár et al., 2017; Pater, 2019; Corkery et al., 2019; McCurdy et al., 2020). Last year, modelling morphological acquisition trajectories was adopted as one of the shared tasks of SIGMORPHON-UniMorph (Kodner and Khalifa, 2022). The three submitted neural systems (Pimentel et al., 2021; Kakolu Ramarao et al., 2022; Elsner and Court, 2022) exhibited over-regularization and developmental regression, but non-human-like behaviours were also observed.

Some recent studies have revealed a poor alignment between the way humans and neural encoder-decoder models generalize to new words (*wug* test) in the case of English verb past tense (Corkery et al., 2019) and German plural nouns (McCurdy et al., 2020). Dankers et al. (2021) observed cognitively plausible representations in a recurrent neural network (RNN) trained to inflect German plural nouns but also found evidence of problematic 'shortcut' learning. Wiemerslage et al. (2022) observed that Transformers resemble humans in learning the morphological inflection of English and German in the *wug* tests but they also pointed out the divergence of the model in German production. However, computational simulations have succeeded in replicating the U-shaped learning curve during the acquisition of past tense (Kirov and Cotterell, 2018; Plunkett and Marchman, 2020). Additionally, further probing experiments have suggested that neural models do learn linguistic representations (Goodwin et al., 2020; Hupkes et al., 2018; Ravichander et al., 2020). Our research continues on exploring the cognitive plausibility of neural networks in modeling language inflection learning.

Recurrent encoder-decoder inflection model In this work, we adopt the model of Kirov and Cotterell (2018), henceforth referred to as **K&C**. This model is based on the encoder-decoder architecture proposed by Bahdanau et al. (2014), with input representation and hyper-parameters taken from Kann and Schütze (2016). The architecture consists of a bidirectional LSTM (BiLSTM) encoder augmented with an attention mechanism and a unidirectional LSTM decoder. The task of the encoder is to map each phonetic (or orthographic) symbol from the input string to a unique embedding and then process that embedding to get a context-sensitive representation of that symbol. The decoder reads the context vector from the final cell of the encoder and generates an output of phoneme/grapheme sequences through training a BiLSTM model with two hidden layers. For

more details on the model, see Bahdanau et al. (2014); Kann and Schütze (2016); Kirov and Cotterell (2018).

3 Datasets

To replicate the results published by K&C, we employ their dataset based on CELEX (Baayen et al., 1993).[2] To extend the experiments to Dutch and German and compare the results to English, we build a new dataset containing past tense forms in all three languages.

3.1 K&C English Dataset

K&C's CELEX-based dataset contains 4,039 English verb types including 3,871 regular verbs and 168 irregular verbs. Each verb is associated with an infinitive form and past tense form, both in International Phonetic Alphabet (IPA). Moreover, each verb is marked as regular or irregular (Albright and Hayes, 2003).

Note that there are label errors in their dataset. For example, `dive-dived`, `dream-dreamed`, `light-lighted` are marked as *irregular*. This is possibly because those verbs have two past tense forms and the other form does not follow the regular inflection (`dive-dove, dream-dreamt, light-light`). However, as the past tense of those verbs in the original dataset aligns with the regular inflection rule of English, we take those verbs as regular ones and manually correct their labels.

3.2 Multilingual Unimorph-based Dataset

We use the morphological annotation dataset Unimorph (McCarthy et al., 2020) as a source of English, Dutch, and German word forms to enable a fair comparison in our multilingual experiments. In this lexicon, each entry consists of the infinitive of the verb, the conjugation, and the tag containing the Part-Of-Speech and inflectional information. Our use of the Unimorph dataset allowed for a wider range of past tense inflection cases compared to the CELEX-based dataset. Unlike the latter, we included more present-past pairs instead of exclusively using infinitive-past pairs. An important adjustment has to be made here because English has only two forms for the present tense (*I/you/we/they*) and only one for the past. By contrast, Dutch and German distinguish more persons

[2]Dataset, code and other experimental details are taken from `https://github.com/ckirov/RevisitPinkerAndPrince`

present(g)	past(g)	present(p)	past(p)	reg
accounts	accounted	@k6nts	@k6ntId	reg
account	accounted	@k6nt	@k6ntId	reg
feels	felt	filz	fElt	irreg
feel	felt	fil	fElt	irreg

(a) English

slaap	sliep	slap	slip	irreg
slaapt	sliep	slapt	slip	irreg
slapen	sliepen	slap@	slip@	irreg
behoef	behoefde	b@huf	b@huvd@	reg
behoeft	behoefde	b@huft	b@huvd@	reg
behoeven	behoefden	b@huv@	b@huvd@	reg

(b) Dutch

berechne	berechnete	b@rExn@	b@rExn@t@	reg
berechnest	berechnetest	b@rExn@st	b@rExn@t@st	reg
berechnet	berechnete	b@rExn@t	b@rExn@t@	reg
berechnen	berechneten	b@rExn@n	b@rExn@t@n	reg
fliehe	floh	fli@	flo	irreg
fliehst	flohst	flist	flost	irreg
flieht	floh	flit	flo	irreg
fliehen	flohen	fli@n	flo@n	irreg

(c) German

Figure 1: Excerpt of the newly introduced dataset of English, Dutch and German past tense. Dutch verbs: `slapen` (*to sleep*); `behoeven` (*to need*). German: `berechnen` (*to calculate*); `fliehen` (*to fleed*).

in both present and past tense. To address this, we include for each lemma the first/second/third singular present form and plural form together with their respective past form, each as a separate entry (see examples in Figure 1).

Specifically, we start by extracting from Unimorph a list of verb lemmas and their corresponding present and past tense forms. A different extraction script is used in each language because of the different number of forms and slightly different POS tags:

- English only has two present tense forms: one for the third person singular and one for the rest. Mostly, there is only one past tense.
- Most verbs in Dutch have three present tense forms and two past tense forms.
- Most verbs in German have five present tense forms and four past tense forms.

Next, we tag each form as regular or irregular, based on a simple rule-based strategy:

- English: if the past tense ends with 'ed' then it is considered a regular verb.
- Dutch: if the singular past tense ends with '-de' or '-te', it is considered regular.

Language	Type	Number of verbs						Total verbs	
		train		dev		test			
		Count	(%)	Count	(%)	Count	(%)	Count	(%)
English	all	4,879	79.9	611	10.0	614	10.1	6,104	100.0
	regular	4,601	75.4	529	8.7	520	8.5	5,650	92.6
	irregular	278	4.6	82	1.3	94	1.5	454	7.4
Dutch	all	4,896	80.1	612	10.0	607	9.9	6,115	100.0
	regular	4,383	71.7	550	9.0	542	8.9	5,475	89.6
	irregular	513	8.4	62	1.0	65	1.0	640	10.4
German	all	4,865	79.7	616	10.1	620	10.2	6,101	100.0
	regular	4,299	70.5	535	8.8	578	9.5	5,412	88.8
	irregular	566	9.2	81	1.3	42	0.7	689	11.2

Table 1: Dataset distributed into train, dev and test sets in each of the three languages. The number of regular and irregular verbs is also reported. The percentage is calculated over the total number of verbs per language.

- German: if the singular past tense of the first or third person ends with '-te', it is considered regular.

Finally, the IPA transcriptions of all word forms are retrieved from CELEX for all languages and added to the final dataset. As shown in Figure 1, the resulting dataset is in the same format as K&C's CELEX-based dataset.

Data selection The generated Dutch data only contains 6106 verb forms *versus* 11489 and 6975 in English and German respectively. Therefore, to enable a fair comparison among languages, we need to downsample the larger datasets. However, randomly choosing 6K verb forms from the English and German lists may lead to a poor selection given the long tail of infrequent words. As a solution, we use word form frequencies as provided in the CELEX data and choose *all* words with a frequency of more than 1 in a million, and complement with a random selection of less frequent words in order to get approximately 6106 verb forms.

To make sure the model can generalize to unseen verbs, we follow Goldman et al. (2022) and split the data by lemma into a train set (80%), a development (dev) set (10%) and a test set (10%). Therefore, the verb forms from the same lemma can only appear in one of the splits. The data distribution into three sets and regular/irregular verbs for each language is reported in Table 1.

3.3 Remarkable problems

A few problems occurred during data preparation. First, rule-based tagging of lemma's is not as trivial as it seems at first sights. For example, in English, not all past tenses ending with '-ed' are regular. Using the data of K&C, we added a few exceptions that are all irregular words ending with '-ed': `bled`, `bred`, `led`, `misled`, `fled`, and forms of `fed` (including `breast-fed`, `force-fed` and `bottle-fed`).

Also, in the original K&C experiment, the model should be able to predict past tense based on what it learned from other verbs, not from other word forms. In morphologically richer languages, a lemma has more word forms and data splitting becomes problematic. For instance, a model might have learned that `work` → `worked` and `walks` → `walked`, then it might predict that `works` → `worked`. In such a case, it is not possible to know whether the model made the right prediction based on similarities to other lemmas (`walks`) or to other forms of the same verb (`work`). To be as comparable as possible to the original setup of K&C, we put all forms of the same verb in the same data split (that is, either training, dev or test). As a result, if the model scores well, we know for sure that it cannot make predictions based on other forms of the same verb.

Another issue is that one present tense form normally corresponds to one past tense form. However, German poses two notable exceptions to this:

- The second person singular verb form ends with '-st' and the third person singular ends with '-t'. Those forms coincide if a verb already ends with an 's', but there is still a difference between those forms in the past tense. For example, `bremst` is the present conju-

gation form of verb `bremsen` (*to brake*) for pronoun `du` *you*, `er` *he* and even `ihr` *you*.

- Verbs ending in '-t' can be the third person singular or the second person plural informal. For example, `wundert` is the present conjugation of the verb `wundern` (*to wonder*) for the pronoun `ihr` *you* and `er` *he*.

In the former case, the model should be able to output multiple solutions, since only context can make clear whether it is the second person or the third person. However, this complicates the evaluation. As a solution, we exclude the third person form if it collides with the second person. As for the latter issue, we choose to remove all second person plural informal forms, since those are far less frequent than the third person singular forms.

4 Replication of K&C

Before moving to the main multilingual experiments, we replicate the original K&C experiments (single-task only).

4.1 Experimental Setup

For the replication, we employ K&C's CELEX-based dataset and keep the model architecture and hyper-parameters unchanged using OpenNMT (Klein et al., 2017)[3]. Also, as reported by K&C, we train the neural model for 100 epochs to make sure the examples in the training data are properly learned. See more details in Appendix A. Following K&C, the model is trained on the IPA transcription.

We use word form-level accuracy to evaluate model performance. An important remark concerns data splitting: K&C did not release their specific data split, which makes it impossible to replicate the exact same results. We, therefore, create our own splits following K&C's proportions (80/10/10% for training/dev/test). To obtain more reliable results, we train the model three times using different random seeds for different initialization and report the averaged resulting accuracies.

To study the micro U-shape learning curve of irregular verbs, we save the model at each 10 epochs and use those partially-trained models to predict the test set and compare their prediction results.

[3]However, as the epoch has been deprecated in the latest version of OpenNMT, we converted it to train_steps based on its relationship with steps.

4.2 Results

As shown in Table 2, the results on the training set are almost the same as reported in the original paper, which means our replication is largely successful.[4] We note that the accuracy for irregular verbs in the dev and test set is considerably different from that of K&C (dev: 21.1% vs. 53.3%; test: 35.3% vs. 28.6%). Since K&C did not release their specific data split, replicating their exact results on the small portion of irregular verbs is not possible. Given that our results are averaged over three random seeds and on all three split sets, we consider them more reliable, which means the model might perform worse at learning the past tense of irregular verbs than K&C's report.

	all			**regular**			**irregular**		
	train	**dev**	**test**	**train**	**dev**	**test**	**train**	**dev**	**test**
K&C	99.8	97.4	95.1	99.9	99.2	98.9	97.6	53.3	28.6
Ours	99.9	95.3	96.5	99.9	98.4	99.2	98.4	21.1	35.3

Table 2: Mean accuracy of our replication of K&C with three random seeds based on English data from CELEX-based dataset.

4.3 Discussion

The reason we assume for the gap between our results and K&C's is twofold: (i) the number of irregular verbs is much lower than regular ones, which makes the accuracy change dramatically even if only few more or few less verbs are predicted correctly than the original experiments; (ii) we corrected the label errors mentioned above, thus the number of irregular verbs becoming smaller than before. This small difference could cause a large impact on the accuracy calculation given that these two sets only contain about 20 irregular verbs. To test this hypothesis, we conduct 9-fold cross-validation[5] and find that the accuracy for irregular verbs varied in different dev splits, ranging widely between 9% and 42%.

[4]Our results are also very close to those of Corkery et al. (2019), who did a similar replication and reported the averaged accuracy over ten runs initialized with different random seeds, but only on the training set.

[5]We keep the test set unchanged and validated across the train and dev sets. To make sure the dev set has a comparable number of verbs as the original set, we adopt 9 fold instead of 10 fold cross-validation.

5 Multilingual Experiments

This section presents the results of our main experiments aimed at comparing Dutch and German past learning patterns to the English ones. It also presents the results of grapheme *vs* phoneme sequence learning in all three languages. Because Dutch and German pronunciation is more predictable than the English one, we expect that the difference between grapheme and phoneme learning will be smaller in these languages.

For comparability, all experiments in this section use the newly introduced Unimorph-based dataset, which includes a similar amount of training forms in all languages (cf. Table 1). The model architecture and the hyperparameter settings are the same as in previous experiments. We also run each experiments three times with different random seeds and report the averaged results.

We use our newly-created data for multilingual experiments without resampling tokens by their frequency. This decision is informed by research suggesting that human learners generalize over type frequency, rather than token frequency (Bybee, 1995; Bybee and Thompson, 1997) and is consistent with the experimental design of K&C. Other studies have suggested that word frequency is important for children's past tense acquisition (Plunkett and Marchman, 1991; Bybee and Slobin, 1982; Ellis, 2002), but we do not examine this hypothesis in this work.

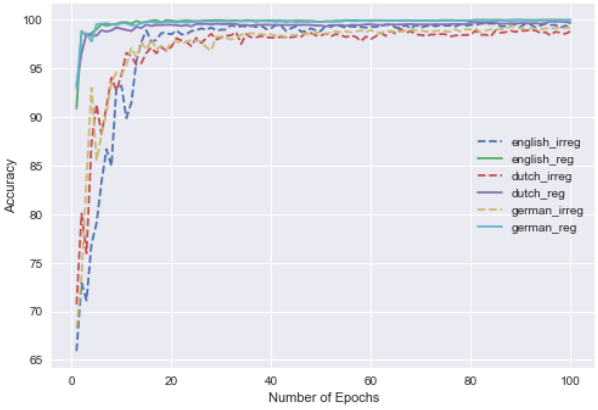
Result overview For the forms seen in training, the model is able to learn both regular and irregular past tense inflection with more than 95% accuracy (Table 3a), and with similar learning curves (Figure 2), which confirms and strengthens the main findings of K&C on two other languages.

Comparing Table 3a to 3b, we find that the overall trends are maintained when the model is trained on graphemes instead of phonemes (the original setup of K&C). However, a notable exception is observed: grapheme learning results in a much lower accuracy of English irregular verbs.

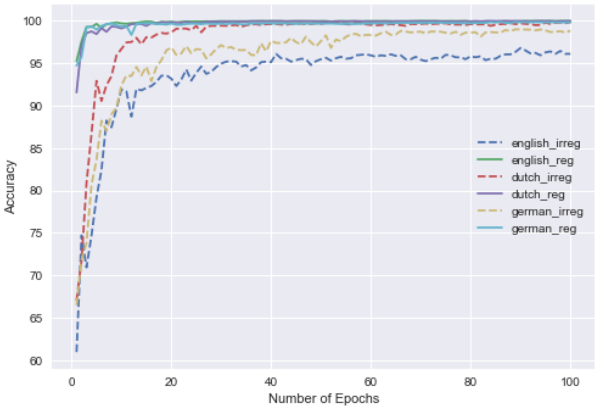
In the following sections, we discuss these results in more detail.

5.1 Past Tense Learning Results in English, Dutch, and German

Accuracy Looking closer at the results across languages (Table 3a), we notice that inflecting *unseen* Dutch regular verbs is slightly harder than in



(a) Phoneme Input



(b) Grapheme Input

Figure 2: Learning curves of the model on the German, English, and Dutch training set (with random seed *123*).

German and English. This might be explained by the fact that in Dutch all voiced consonants become unvoiced at the end of a word, but to predict if the past tense becomes '-de' (for voiced consonants) or '-te' (for unvoiced consonants), we still need the end consonant of the stem, which can be found within the lemma and most of the times in the spelling of the word form. Unfortunately, this information is absent in the pronunciation. For example, in the pair `lAnt-lAndd@`, one will not know whether the past tense should be `lAnd@` or `lAnt@` before seeing the orthographic form `land`. We find that such errors account for about 50% (18/38) of all Dutch regular verb errors. This difference in voiced/unvoiced regular past tense endings only occurs in Dutch.

As for irregular verbs, we find a large difference across languages in the ability to generalize to new forms. Especially in English, while the model has almost perfectly learned to inflect seen verbs, it has a hard time predicting the form of new irreg-

	all			regular			irregular		
	train	**dev**	**test**	**train**	**dev**	**test**	**train**	**dev**	**test**
EN	99.5	93.1	92.1	99.8	96.1	95.0	98.1	27.8	40.5
NL	98.9	88.4	88.4	99.2	91.4	92.2	96.5	62.4	57.9
DE	98.9	85.0	92.5	99.4	92.0	95.1	96.7	38.7	57.9

(a) Phoneme input

	all			regular			irregular		
	train	**dev**	**test**	**train**	**dev**	**test**	**train**	**dev**	**test**
EN	99.1	93.6	93.8	99.8	98.2	98.1	89.0	11.1	28.1
NL	99.4	88.0	89.6	99.8	91.2	93.0	97.9	58.6	61.0
DE	98.4	86.4	93.6	99.1	93.5	95.7	93.9	39.5	65.9

(b) Grapheme input

Table 3: Past tense inflection accuracy in English, Dutch, and German; all averaged over 3 random seeds.

epoch	**English**		**Dutch**		**German**	
	hits		**bestijgt** (*mounts*)		**gilt** (*applies*)	
10	hItId	hitted	b@stKGd@	besteeg	gIlt@	galte
20	hItst	**hit**	**b@stex**	besteeg	gIlt@	**galt**
30	hItId	**hitted**	**b@stKGd@**	besteeg	**g&lt**	galt
40	hItId	hitted	b@stKGd@	besteeg	g&lt	galt
50	**hIt**	hitted	b@stKGd@	besteeg	g&lt	galt
60	**hItst**	**hit**	**b@stex**	besteeg	**gIlt@**	**gilte**
70	**hIt**	hit	b@stex	**bestijgde**	**g&lt**	**galt**
80	**hItId**	**hitted**	b@stex	**besteeg**	g&lt	galt
90	hItId	hitted	b@stex	besteeg	g&lt	galt
100	hIt	hit	b@stex	besteeg	g&lt	galt

Table 4: The oscillating development (micro U-shape) of single verbs in three languages: with phoneme or grapheme inputs, the respectively predicted past phonetic (left) or orthographic (right) forms are changing with the training proceeding, but their final predictions are correct when reaching the last epoch. The changing points are boldfaced.

ular verbs (dev: 27.8%, test: 40.5%). This effect is smaller in Dutch and German, suggesting the irregular inflection patterns in these languages are more predictable. Surprisingly, the model made more mistakes when predicting the inflections of the irregular verbs in the German dev set than the test set (dev: 38.7%, test: 57.9%). By inspecting the mistakes, we found that the model incorrectly took many irregular verbs as regular ones because of their resemblance (high character overlap). For instance, `reitest–⋆reitetest/rittest` (*ride*) is influenced by the regular conjugation of `bereitest–bereitetest` (*prepare*). We found 23/81 irregular verbs in the dev set are very similar to regular verbs in the training set. Out of these, 8 irregular verbs are identical to regular ones except for a prefix (e.g., `reitet` (*rides*) vs. `bereitet` (*prepares*) and `reitest` (*ride*) vs. `verbreitest` (*spread*), which could be highly confusing for a model that is only based on form regardless of meaning. By contrast, such overlap is not found between the irregular verbs in the test set and regular ones in the training set. This distributional discrepancy might explain the lower accuracy in the dev set. It echoes with our other finding discussed in the next section that irregular verbs might be misled by regular verbs if they share representation similarity.

Errors and learning trajectories Going beyond overall accuracy, we inspect the learning trajectories of individual verbs in our dataset. We find human-like overregularization patterns similar to those observed by K&C in English also occur in Dutch and German. For example, in Dutch, after 40 epochs of training, the model change `verscheent` to `verscheen` as the past tense of `verschijnt` (*appears*). However, after 50 epochs, the model again generate the wrong form `verscheent`. After 70 epochs, the correct result is again obtained. Similar patterns are observed for `sink` in English and `streitet` (*argues*) in German. Interestingly, Plunkett and Marchman (1991); Bybee and Slobin (1982); Kuczaj II (1977) reported that children do sometimes vacillate, even within one utterance, between the correct and incorrect past tense form of the same irregular stem. All wrongly predicted irregular verbs are caused by over-regularization. In other words, no patterns like `ated` in English or `lookte` in Dutch are

found, which is consistent with humans' learning behaviour (Pinker and Prince, 1988). More examples from English, Dutch and German are listed in Table 4.

Additionally, we find cases where the model generates an irregular form for a regular verb, because of the resemblance with other (irregular) verbs. In Dutch, for example, the regular verb `versier-versierde` (*decorate-decorated*) gets incorrectly inflected as `*versoor` by resemblance to verbs like `verlies-verloor` (*lose-lost*). Similar errors also occur in German. For instance, the wrong prediction of `verfehle-*verfahl/verfehlte` (*miss-missed*) might be misled by the pair `befehlen-befahlen` (*order-ordered*), and `schweben-*schwoben/schwebten` (*float-floated*) is possibly due to its resemblance to `schieben-schoben` (*push-pushed*). Interestingly, this type of errors aligns with Ernestus and Baayen (2004)'s experiments with Dutch speakers: phonological similarity, rather than rule-based regularity, influences participants' judgments toward the inflection of verbs.

That said, the model also displays error patterns that are *not* human-like, such as copying the present form or randomly removing phonemes (or letters) from it. Similar cases of non-plausible predictions were also observed at the Sigmorphon Shared Task (Kodner and Khalifa, 2022), for instance `forgive-*forgaved/forgave` or `seek-*sougk/sought`. As also observed by Wiemerslage et al. (2022), this kind of model predictions contrasts with the behavior of human speakers, who mostly resort to generating a regular past tense when a verb is unknown.

5.2 Phoneme vs. Grapheme Input

Undoubtedly, using phoneme input is more principled than grapheme input when simulating human acquisition patterns. However, pronunciation information is not always available and makes it harder to extend this kind of simulations beyond a small set of widely studied languages. Here, we investigate the usability of grapheme-based input for modeling past tense inflection. We expect German and Dutch to be a good use case for this, given their more transparent orthography compared to English (Marjou, 2021).

The results in Table 3 clearly show that switching to grapheme input for the English simulations is not principled as this results in a slight *increase* of regular inflection accuracy (from 99.8/96.1/95.0% to 99.8/98.2/98.1% train/dev/test) as opposed to a large *decrease* of irregular inflection accuracy (from 98.1/27.8/40.5% to 89.0/11.1/28.1%). The latter effect is particularly marked, suggesting non-transparent orthography may not be a uniform property of the language but may be correlating with less regular word forms within a language. We leave this investigation to future work.

Using grapheme input in Dutch and German seems much safer (differences are overall small, with only a slight increase in almost all cases). Our observations seem to reflect the figures of Marjou (2021), who give a much higher transparency score to Dutch and German than to English.

In sum, using graphemes to simulate human patterns of morphological acquisition is possible but should be done with caution and only in some languages. A good practice could be to first verify that the orthographic transparency of a language is high (Marjou (2021) present results for 17 languages). When that is not possible, grapheme-based results should be at least validated against a small-scale pronunciation dataset.

6 Conclusions

In this work, we study the plausibility of using sequence-to-sequence neural networks for simulating human patterns of past tense acquisition. More specifically, we replicate findings by Kirov and Cotterell (2018) and examine their generalizability beyond the specific case of English, using a new dataset of English/Dutch/German (ir)regular verb forms based on Unimorph (McCarthy et al., 2020).

We show that the main findings of K&C also largely hold for Dutch and German, including over-regularization errors and the oscillating (or micro U-shape) learning trajectory of individual verb forms across training epochs. At the same time, we also observe cases of non human-like errors, for instance when the model just keeps the present form unchanged or randomly removes phonemes from it. A notable difference among our studied languages concern unseen English irregular verbs, which appear to be much harder to inflect than the Dutch and German ones. We also observe that the orthographic transparency of a language influences and possibly confounds the

model's learning performance: higher transparent orthography contributes to more reliable and consistent simulation results, but in general this aspect should be seriously considered when setting up new benchmarks of morphological acquisition.

Future work could include the construction of a nonce word benchmark in Dutch and German to enable a multi-lingual evaluation of this task (Corkery et al., 2019), as well as an in-depth investigation of the different level of irregular past inflection difficulty in our three languages.

Kirov and Cotterell (2018) provided very promising evidence for the use of modern neural networks to model the human language acquisition patterns. Our work confirms the potential of this research direction, but also raises important issues and joins recent follow-up studies (Corkery et al., 2019; Dankers et al., 2021; Kodner and Khalifa, 2022; Wiemerslage et al., 2022) that have warned against over-optimistic conclusions.

References

Adam Albright and Bruce Hayes. 2003. Rules vs. analogy in english past tenses: A computational/experimental study. *Cognition*, 90(2):119–161.

R Harald Baayen, Richard Piepenbrock, and H Van Rijn. 1993. The celex lexical database (cd-rom). linguistic data consortium. *Philadelphia, PA: University of Pennsylvania*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ryan P Blything, Ben Ambridge, and Elena VM Lieven. 2018. Children's acquisition of the english past-tense: Evidence for a single-route account from novel verb production data. *Cognitive Science*, 42:621–639.

A Van den Bosch, Alain Content, W Daelemans, and Béatrice De Gelder. 1994. Analysing orthographic depth of different languages using data-oriented algorithms: Qualico94. In *Proceedings of the 2d International Conference on Quantitative Linguistics*, pages 26–31.

Joan Bybee. 1995. Regular morphology and the lexicon. *Language and cognitive processes*, 10(5):425–455.

Joan Bybee and Sandra Thompson. 1997. Three frequency effects in syntax. In *Annual Meeting of the Berkeley Linguistics Society*, volume 23, pages 378–388.

Joan L Bybee and Dan I Slobin. 1982. Rules and schemas in the development and use of the english past tense. *Language*, 58(2):265–289.

Maria Corkery, Yevgen Matusevych, and Sharon Goldwater. 2019. Are we there yet? encoder-decoder neural networks as cognitive models of English past tense inflection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3868–3877, Florence, Italy. Association for Computational Linguistics.

Verna Dankers, Anna Langedijk, Kate McCurdy, Adina Williams, and Dieuwke Hupkes. 2021. Generalising to German plural noun classes, from the perspective of a recurrent neural network. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 94–108, Online. Association for Computational Linguistics.

Nick C Ellis. 2002. Frequency effects in language processing: A review with implications for theories of implicit and explicit language acquisition. *Studies in second language acquisition*, 24(2):143–188.

Micha Elsner and Sara Court. 2022. OSU at SigMorphon 2022: Analogical inflection with rule features. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 220–225, Seattle, Washington. Association for Computational Linguistics.

Mirjam Ernestus and Harald Baayen. 2004. Analogical effects in regular past tense production in dutch.

Omer Goldman, David Guriel, and Reut Tsarfaty. 2022. (un)solving morphological inflection: Lemma overlap artificially inflates models' performance. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 864–870, Dublin, Ireland. Association for Computational Linguistics.

Emily Goodwin, Koustuv Sinha, and Timothy J. O'Donnell. 2020. Probing linguistic systematicity. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1958–1969, Online. Association for Computational Linguistics.

Ulrike Hahn and Ramin Charles Nakisa. 2000. German inflection: Single route or dual route? *Cognitive Psychology*, 41(4):313–360.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Akos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2017. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics*, 43(4):761–780.

Akhilesh Kakolu Ramarao, Yulia Zinova, Kevin Tang, and Ruben van de Vijver. 2022. HeiMorph at SIGMORPHON 2022 shared task on morphological acquisition trajectories. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 236–239, Seattle, Washington. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2016. Med: The lmu system for the sigmorphon 2016 shared task on morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 62–70.

Christo Kirov and Ryan Cotterell. 2018. Recurrent neural networks in linguistic theory: Revisiting pinker and prince (1988) and the past tense debate. *Transactions of the Association for Computational Linguistics*, 6:651–665.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Jordan Kodner and Salam Khalifa. 2022. SIGMORPHON–UniMorph 2022 shared task 0: Modeling inflection in language acquisition. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 157–175, Seattle, Washington. Association for Computational Linguistics.

Stan A Kuczaj II. 1977. The acquisition of regular and irregular past tense forms. *Journal of verbal learning and verbal behavior*, 16(5):589–600.

Gary F Marcus, Ursula Brinkmann, Harald Clahsen, Richard Wiese, and Steven Pinker. 1995. German inflection: The exception that proves the rule. *Cognitive psychology*, 29(3):189–256.

Gary F Marcus, Steven Pinker, Michael Ullman, Michelle Hollander, T John Rosen, Fei Xu, and Harald Clahsen. 1992. Overregularization in language acquisition. *Monographs of the society for research in child development*, pages i–178.

Xavier Marjou. 2021. OTEANN: Estimating the transparency of orthographies with an artificial neural network. In *Proceedings of the Third Workshop on Computational Typology and Multilingual NLP*, pages 1–9, Online. Association for Computational Linguistics.

Arya D. McCarthy, Christo Kirov, Matteo Grella, Amrit Nidhi, Patrick Xia, Kyle Gorman, Ekaterina Vylomova, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, Timofey Arkhangelskiy, Nataly Krizhanovsky, Andrew Krizhanovsky, Elena Klyachko, Alexey Sorokin, John Mansfield, Valts Ernštreits, Yuval Pinter, Cassandra L. Jacobs, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2020. UniMorph 3.0: Universal Morphology. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3922–3931, Marseille, France. European Language Resources Association.

Kate McCurdy, Sharon Goldwater, and Adam Lopez. 2020. Inflecting when there's no majority: Limitations of encoder-decoder neural networks as cognitive models for German plurals. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1745–1756, Online. Association for Computational Linguistics.

Joe Pater. 2019. Generative linguistics and neural networks at 60: Foundation, friction, and fusion. *Language*, 95(1):e41–e74.

Tiago Pimentel, Maria Ryskina, Sabrina J. Mielke, Shijie Wu, Eleanor Chodroff, Brian Leonard, Garrett Nicolai, Yustinus Ghanggo Ate, Salam Khalifa, Nizar Habash, Charbel El-Khaissi, Omer Goldman, Michael Gasser, William Lane, Matt Coler, Arturo Oncevay, Jaime Rafael Montoya Samame, Gema Celeste Silva Villegas, Adam Ek, Jean-Philippe Bernardy, Andrey Shcherbakov, Aziyana Bayyr-ool, Karina Sheifer, Sofya Ganieva, Matvey Plugaryov, Elena Klyachko, Ali Salehi, Andrew Krizhanovsky, Natalia Krizhanovsky, Clara Vania, Sardana Ivanova, Aelita Salchak, Christopher Straughn, Zoey Liu, Jonathan North Washington, Duygu Ataman, Witold Kieraś, Marcin Woliński, Totok Suhardijanto, Niklas Stoehr, Zahroh Nuriah, Shyam Ratan, Francis M. Tyers, Edoardo M. Ponti, Grant Aiton, Richard J. Hatcher, Emily Prud'hommeaux, Ritesh Kumar, Mans Hulden, Botond Barta, Dorina Lakatos, Gábor Szolnok, Judit Ács, Mohit Raj, David Yarowsky, Ryan Cotterell, Ben Ambridge, and Ekaterina Vylomova. 2021. SIGMORPHON 2021 shared task on morphological reinflection: Generalization across languages. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–259, Online. Association for Computational Linguistics.

Steven Pinker and Alan Prince. 1988. On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28(1-2):73–193.

Steven Pinker and Michael T Ullman. 2002. The past and future of the past tense. *Trends in cognitive sciences*, 6(11):456–463.

David C Plaut and Laura M Gonnerman. 2000. Are non-semantic morphological effects incompatible with a distributed connectionist approach to lexical processing? *Language and Cognitive Processes*, 15(4-5):445–485.

Kim Plunkett and Patrick Juola. 1999. A connectionist model of english past tense and plural morphology. *Cognitive Science*, 23(4):463–490.

Kim Plunkett and Virginia Marchman. 1991. U-shaped learning and frequency effects in a multi-layered perception: Implications for child language acquisition. *Cognition*, 38(1):43–102.

Kim Plunkett and Virginia Marchman. 2020. U-shaped learning and frequency effects in a multilayered perceptron: Implications for child language acquisition. *Connectionist psychology: A text with readings*, pages 487–526.

Kim Plunkett, Virginia Marchman, and Steen Ladegaard Knudsen. 1991. From rote learning to system building: acquiring verb morphology in children and connectionist nets. In *Connectionist Models*, pages 201–219. Elsevier.

Michael Ramscar. 2002. The role of meaning in inflection: Why the past tense does not require a rule. *Cognitive Psychology*, 45(1):45–94.

Abhilasha Ravichander, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. On the systematicity of probing contextualized word representations: The case of hypernymy in bert. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 88–102.

David E Rumelhart and James L McClelland. 1986. On learning the past tenses of english verbs.

Mark S Seidenberg and Laura M Gonnerman. 2000. Explaining derivational morphology as the convergence of codes. *Trends in cognitive sciences*, 4(9):353–361.

Niels A Taatgen and John R Anderson. 2002. Why do children learn to say "broke"? a model of learning the past tense without feedback. *Cognition*, 86(2):123–155.

Adam Wiemerslage, Shiran Dudy, and Katharina Kann. 2022. A comprehensive comparison of neural networks as cognitive models of inflection. *arXiv preprint arXiv:2210.12321*.

A Appendix

Parameter	Value
seed	123
feat_vec_size	300
feat_merge	concat
rnn_type	LSTM
encoder_type	brnn
encoder_layers	2
encoder_rnn_size	100
decoder_type	rnn
decoder_layers	2
decoder_rnn_size	100
dropout	0.3
learning_rate_decay	1.0
learning_rate	1.0
batch_size	20
train_steps	(training sample size/ batch size)∗the number of epochs
beam_size	12
optim	adadelta
verbose	True
tensorboard	True
tensorboard_log_dir	logs
report_every	steps / 100
log_file	directory of the log file
log_file_level	20

A displays hyperparameter settings of the replicating experiments and the extension experiments.

Class Explanations: the Role of Domain-Specific Content and Stop Words

Denitsa Saynova[1], **Bastiaan Bruinsma**[1], **Moa Johansson**[1], **Richard Johansson**[1,2]
[1]Chalmers University of Technology, Gothenburg, Sweden
[2]University of Gothenburg, Gothenburg, Sweden
{saynova, sebastianus.bruinsma, moa.johansson}@chalmers.se
richard.johansson@cse.gu.se

Abstract

We address two understudied areas related to explainability for neural text models. First, *class explanations*. What features are descriptive across a class, rather than explaining single input instances? Second, the *type of features* that are used for providing explanations. Does the explanation involve the statistical pattern of word usage or the presence of domain-specific content words? Here, we present a method to extract both class explanations and strategies to differentiate between two types of explanations – domain-specific signals or statistical variations in frequencies of common words. We demonstrate our method using a case study in which we analyse transcripts of political debates in the Swedish Riksdag.

1 Introduction

Recent developments in NLP are often the result of ever more complex model architectures and an increasing number of model parameters. Yet, if we want to rely on these models, we should be able to review the similarities and dissimilarities between the model and human judgement. Explainability frameworks can do this by highlighting on *what* the model has learnt to base its decisions. Are these coincidental statistical patterns or something that a human would use as an explanation? Madsen et al. (2022) argue that explanations should ideally be both *functionally-grounded* (true to the underlying machine learning model) as well as *human-grounded* (useful to a human).

In this article, we propose a new method for extracting class explanations from text classifiers. Besides, we also show a new way to distinguish between two types of features that appear in those explanations, that is, between informative content words and subtle statistical differences in common words' frequencies. Our method aggregates explanations for individual data points (here provided by LIME (Ribeiro et al., 2016)), followed by a sorting stage that separates the different kinds of features.

Our work is in part motivated by use cases of machine learning for texts in the social sciences. In this field, explainability methods are relevant both as checks to compare against human expert knowledge and as a tool for bias detection. As a case study, we use our method to explain the decisions of a binary classifier trained to identify if speeches in the Swedish Riksdag belong to either of the two main parties, the Moderates (M) or the Social Democrats (S).

We find that our method can separate class explainability features and that those data points whose explanations contain primarily domain-specific content words are more often classified correctly.

2 Literature Review

As a result of the extensive work on explainability methods, a complex typology of different approaches exists (see Danilevsky et al. (2020) or Madsen et al. (2022) for a survey). One important distinction is between *global* and *local*. On the one hand, global methods aim to explain some general behaviour of a model, such as class explanations, which summarise the model with respect to a certain class. On the other, local methods aim to explain why the model assigned a single data point to a particular class.

Between global and local methods, the latter receive the most attention (Nauta et al., 2022). Three popular methods are gradient-based approaches (Baehrens et al., 2010), Shapley values (Shapley, 1952), and LIME. Gradient-based approaches use the model's weights and take the gradient with regard to the input. As such, they measure the

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 103–112

change in the outcome given some small change in the input. Yet, they are only an accurate reflection of the model if that model is linear (Li et al., 2016), which is not the case for most deep NLP architectures. On the other hand, while Shapley values have many theoretical guarantees to make them a faithful interpretation (they represent the true contributions of the features (Ethayarajh and Jurafsky, 2021)), their implementations (e.g. via attention flows for transformer-based architectures (Abnar and Zuidema, 2020)) tend to be computationally expensive, which is problematic in the current setting, where we focus on aggregating a substantial number of individual explanations. Finally, LIME has an advantage over gradient-based approaches as it is model agnostic. This means that LIME attempts to explain a trained classifier independently of its architecture (Ribeiro et al., 2016).

2.1 Class explanations

The area of global *class explanations* is so far less studied than that of local explanations. One approach to providing global understanding of the model is to use behavioural or structural probes (Tenney et al., 2019; Hewitt and Manning, 2019; Wallace et al., 2019). Probing is a technique where a supervised model (a probe) is used to determine what is encoded in the internal representation of the studied model. This is done by training the probe to predict based on the frozen representations of the black-box model. If the probe performs well on the task, that indicates the required information was well represented by the black-box model, if the probe is unable to achieve high accuracy, that is taken to signify that the studied patterns are not learned by the black-box model. This has some limitations – for example, the complexity of the probe. If the probe is too simple, it may not capture second order effects, if it is too complex, it may learn the task internally and "discover" things that are in the probe rather than the model (Hewitt and Liang, 2019). More importantly, these methods tend to be applied to the discovery of simple syntactic structures like part of speech (POS) tagging, syntactic tree structures (Rogers et al., 2020) or to detect the presence of specific knowledge (Petroni et al., 2019). Other attempts in this area include leveraging local methods and utilising a strategy for aggregating and presenting those results to the user. An example of such approach is SP-LIME (Ribeiro et al., 2016), which aggregates individual LIME explanations with a greedy search for finding data points (texts) that are explained by the most dissimilar sets of features in order to represent the breadth of the class explanations. The results are presented as ranked text examples with their corresponding explanations, where the number of examples is defined by the user. Due to its focus on features that cover as many input instances as possible, this method tends to overemphasise stop words (see further discussion in Section 6).

2.2 Features of Explanations

To a human, not all features learnt by the machine learning model are equally informative. Some signals may come from speech patterns, others from the topic that is discussed and the sentiment, yet others may indicate preferred catchphrases and slogans. There is a distinction between explanations of the model (what a model bases its prediction on) and human explanation (what a human would base their decision on if faced with the same prediction task) (Miller, 2019). Since humans have background knowledge that is not accessible to the model and the model has the capacity to detect small statistical signals that are beyond human computational capabilities, the set of features that are selected by either may differ. This issue can be viewed in terms of the concepts presented in the position paper by Doshi-Velez and Kim (2017) and further discussed by Madsen et al. (2022), namely – *human-grounded* and *functionally-grounded* explainability. Functionally-grounded explainability is concerned with how well the explanation reflects the model, whereas human-grounded explainability is concerned with producing explanations that are useful to a human. This is also in line with work by Nauta et al. (2022), where the authors argue for the rigorous evaluation of an explainability method across twelve properties in three categories – content, presentation, and user. The content properties and in particular *correctness* (faithfulness w.r.t. the black box) are related to the functionally-grounded approach, whereas the user properties – *context* (how relevant the explanation is to the user), *coherence* (how accordant the explanation is with prior knowledge), and *controllability* (how interactive or controllable an explanation is) – relate to human-grounded explainability.

In our work, we use stop words and content

words to align with functionally-grounded and human-grounded explanations. *Content words* are words that have independent meaning outside of the sentence they appear in. These are typically a noun, verb, adjective, or adverb and are distinguished from *function words*, which mainly express grammatical relationships and have little semantic content. *Stop words* are words that carry little or no important information for the task at hand and tend to contain *function words*. This concept is not strictly defined, but generally refers to high-frequency terms. It can therefore extend to, for example, procedural language (e.g. "tallman" (speaker)) that can also act as a stop word in the domain of Swedish political debates. A model can learn to detect distributional differences of any word as long as it is correlated with the predicted class, but a human will be unlikely to relate and understand the cause of the distributional differences of stop words. The difference in frequency of how often a group uses the word "also", for example, may not be very informative for a human, even if those distributional differences point to real speech patterns that distinguish between the speakers (Arun et al., 2009a) and have even been linked to the author's gender (Arun et al., 2009b). Human domain knowledge will most likely be captured through domain-specific, content words. Being able to confirm the (extent of the) model's grounding in content words can serve to validate it.

3 Method

Our algorithm for computing class explanations consists of four steps: post-hoc instance explanations extraction, aggregation, sorting, and a keyword-in-context search that extracts example texts. This framework is formalized in Algorithm 1. It is similar to SP-LIME, but rather than searching for data points that capture the most diversity of the important features, we propose to work directly with the feature importance and explore ways to summarize and sort these by relevance.

The replication materials and full results are available online [1].

3.1 Step 1: Instance explanation extraction

For a set of held-out data samples N, we apply the trained classifier f. In the instances where the classifier makes the correct prediction, we extract the list of features and their corresponding saliency with model g. This can also be flipped to focus on instances where the model makes the incorrect predictions to investigate which patterns or instances are hard to classify. A certainty threshold can also be used to explore only cases where the model is certain or borderline cases. Our method aims to be extendable to different model architectures, therefore we require a post-hoc, model agnostic instance explanation function g. For now, we have chosen LIME, but alternative

Algorithm 1 Class explainability from instance explanations

Require: Binary classifier f, data samples N
Require: Instance explainability function g
Require: Feature scoring function h
$W \leftarrow \{\}$ ▷ features and importance scores
$c1 \leftarrow \{\}$ ▷ features explaining class 1
$c2 \leftarrow \{\}$ ▷ features explaining class 2

Step 1 – Instance explanation extraction

for $text, true\_label \in N$ **do**
  if $f(text) = true\_label$ **then**
    $W \leftarrow W \cup \{g(text, f)\}$
  end if
end for

Step 2 – Aggregation

for $feature, score \in W$ **do**
  if $score < 0$ **then**
    $c1 \leftarrow c1 \cup \{feature\}$
  else
    $c2 \leftarrow c2 \cup \{feature\}$
  end if
end for

Step 3 – Sorting

for $c \in \{c1, c2\}$ **do**
  return c sorted by h score
end for

Step 4 – Keywords in context

for $c \in \{c1, c2\}$ **do**
  for $term \in$ top X terms in c **do**
    return all occurrences of $term$
    with n words before and after
  end for
end for

[1] https://github.com/dsaynova/NoDaLiDa2023

methods can be used as well, as long as they are able to extract features and the feature contribution scores that explain an instance. This means we are currently constrained by LIME's limitations and only consider single tokens as features. Since LIME is a surrogate model, there is also some uncoupling between the classification model and the explanations. For each correctly classified instance, we extract the top k features (here set to 10). This can be reduced even further in order to limit the number of features that are considered or extended to include all tokens and the task of limiting the explanation will then be completely relegated to the sorting step.

3.2 Step 2: Aggregation

A feature can contribute either positively or negatively towards the prediction of the model. When working with a binary classifier, a negatively contributing feature towards predicting class 1 means it is a positively contributing feature for class 2. Therefore, the features collected from the previous step are aggregated in two sets – $c1$, $c2$ – one for each class based on their feature score sign. Note that these two sets of features may have overlaps if the predictive signal is indicative of the different context in which those features appear.

3.3 Step 3: Sorting

The resulting sets of features for each class need to be constrained to a feasible size to be interpretable by a human. We propose two approaches to developing a feature relevance score h to prioritize and distinguish these terms along an axis of more domain-specific concepts to more generic words – *normalization* and *PCA*.

Normalization. Here, we use the sum of LIME scores for each feature of the explanation divided by number of occurrences of that feature in the validation set. We calculate the feature relevance score h of the j^{th} feature as: $h_j = \frac{1}{m_j} \sum_{i=1}^{N} W_{ij}$. Here, N is the number of data points in the explained dataset, m_j is the number of occurrences of feature j in the explained set, and W is the explanation matrix containing the local importance of the interpretable components for each instance. This will give higher scores to features identified as more important by LIME, but will penalise common words, if they do not contribute to a class prediction often. This is in line with the definition of stop words and should target the corpus-specific stop words. We also filter out words that appear in two or less documents, as these can be party specific, but may not be useful for generalisation. This number can also be increased to filter out more predictive (according to LIME) words.

PCA. The second approach to sorting is to decouple it from the LIME score after the initial aggregation step and use PCA of word embeddings. We found that PCA applied to pre-trained word embeddings tends to separate domain specific words from more generic terms. A theoretical motivation for this analysis lies in the distributional differences between a general text (used for pre-training word embeddings) and a domain-specific text (in this case – political debate). We hypothesise that the general embedding model will see the domain specific terms in sufficiently distinct context in order to embed them in a compact space with a latent dimension separating them from more common and general terms. This relies on the studied data having a significant amount of domain specific terminology that is rarer in general. We expect this to be the case for many application within the social sciences (e.g. politics), but can have limitations in, lower-level, syntactic classification tasks like POS tagging.

To calculate the sorting score, the terms from each set $c1$ and $c2$ are embedded using a model[2] trained on the Swedish CoNLL17 corpus. A PCA is run on each set of words – $c1$, $c2$ – and the first PCA dimension value is used as the sorting score h. Similarly to the normalisation approach, words that appear in two or fewer documents are filtered out. This dimension seems to provide a good distinction of domain specific terms.

3.4 Step 4: Keywords in Context

To further increase human interpretability, we also provide a way to provide context by extracting snippets of texts around the top word features produced in Step 3. For each occurrence, we use a simple keyword-in-context search and extract n words before and after our feature word. This is clearly not feasible or interesting for very frequent words, which further motivates separating rarer, domain specific content words from more common stop words.

[2] http://vectors.nlpl.eu/repository/20/69.zip

4 Data

The dataset used for the case-study consists of transcripts of debates in the Swedish Riksdag, sourced from Riksdagens öppna data – Anföranden[3]. We use a pre-processed version available from Språkbanken[4] consisting of debates from 1993 to 2018. For our experiment, texts from the Social Democrat (S) and Moderate (M) parties have been extracted, resulting in 104,842 S and 62,160 M data points (one data point is one speech that could be part of a longer debate). From these, 100 examples have been sampled for a small-scale human baseline check, where two annotators are asked to perform the classification task of determining the party label from the speech texts and were evaluated against the true label. Since these are debates, references to the opponent are a strong but trivial predictor of party. References to people and political parties have been removed by targeting Swedish political party names' stems (for a full list please refer to the linked code base) and words tagged as "People_along_political_spectrum" in Språkbanken's tags, based on Swedish FrameNet (Heppin and Gronostaj, 2012). Since the cleanup is based on a coarse rule for party name stems detection and the automatic tags from Språkbanken, not all references have been removed. We have opted for blanking all certain cases, so that enough of the interfering signal is removed to make the classification task non-trivial, rather than applying a comprehensive and exhaustive search of all mentions, since that is not the main goal of this work. Data points shorter than 50 words have been removed, as manual analysis shows these tend to be entirely procedural and do not carry political sentiment. This is in line with similar cleaning practices used for US congressional debates (Bayram et al., 2019). The data is undersampled to balance the classes and split into: train (108,169), test (12,019) and validation (2,000) sets. The validation set is used for explainability methods.

5 Experiments

To test our methodology we apply it to a BERT classifier trained to predict the party label of a text (Devlin et al., 2019). The classifier is fine-tuned from a pre-trained model for Swedish data released by The National Library of Sweden/KBLab and available through the huggingface library[5]. The model has a 50,325 word vocabulary and 512 maximum token length. Longer inputs are truncated. As a baseline for investigating class differences and separability of the data we use a logistic regression classifier, as this provides easy access to class explanations by simply looking at the top and bottom scoring internal weights of the model. N-gram spans from 1 to 3 and a combination of all have been compared. The number of input features is 50,325 – the same as the pre-trained BERT model.

A small-scale human annotation check on 100 instances shows the two annotators perform with 58 and 56 percent accuracy respectively. A Cohen's kappa of 0.4 indicates this is a hard classification task.

In the interest of space, the sections below contain partial results. The full results are available online.

5.1 Baseline

Table 1 summarises the accuracy and F1 scores for the logistic regression classifier. We observe that the best result is achieved with 1-grams, with the inclusion of 2- and 3- grams adding no performance gains. It seems the main part of the distinguishing signal can be picked up by specific words rather than phrases.

n-gram span	# feat	acc	F1
1,1	50,325	76.94	76.80
2,2	50,325	73.19	73.05
3,3	50,325	69.39	69.15
1,3	150,975	76.93	76.80

Table 1: Logistic regression classifier performance.

From the internal model weights, we can identify both domain specific words – "sjuka" (sick), "arbetslösa" (unemployed), "arbetslinjen" (the employment line, a Moderate catchphrase), and stop words – "det" (the), "också" (also), "synnerhet" (in particular), can be predictive of the party label. This is in agreement with our assumption that a model can depend on both statistical differences in stop word or in human concepts as

[3] https://data.riksdagen.se/data/anforanden/

[4] https://spraakbanken.gu.se/resurser/rd-anf-1993-2018

[5] https://huggingface.co/KB/bert-base-swedish-cased

the basis of its prediction, and in doing so outperforms the human annotators.

5.2 BERT

The BERT model (lr = 5e-6, batch size = 48, steps = 6000) shows only slight improvement over the baseline, summarised in Table 2.

Evaluation	acc	F1
test set	78.44	76.66
validation set	79.95	78.27

Table 2: BERT classifier performance.

Applying LIME to all validation samples and aggregating the top 10 features for each data point results is a list of 2,043 Moderate and 2,085 Social Democrats terms. Out of these 1,456 Moderate and 1,334 Social Democrat terms appear in more than two documents, and are thus candidates to be included as part of class explanations (this limit can be adjusted by the user).

5.3 Validation

Tables 3 – 4 show the results of both LIME and PCA for both M and S. In both cases, the models separate informative terms from generic ones. This is especially the case with the LIME scores, where the lowest-scoring words are all stop words. As for the highest-scoring words, we find that they are all related to taxes and employment. This is understandable, as this is also what makes up the main political left/right dimension in Sweden (Franzmann and Kaiser, 2006; Jolly et al., 2022; Ezrow et al., 2011). Besides, we can identify several references to several (groups of) parties and ministers, which we would expect in debates. As discussed in section 3.2, we also find a term that appears as important for both parties - *budgetpropositionen* (the budget bill). This is a result of the explainability model using single tokens as features and most likely indicates that this is a term mentioned in a different context for both parties.

While these findings are hopeful on their own, to be useful for social scientists, we need to do

PCA ordering	
rank	**term**
1	utgiftsområde (expenditure area)
2	budgetpropositionen (the budget bill)
3	jobbskatteavdrag (employment tax credit)
4	arbetslöshetsförsäkringen (unemployment insurance)
5	skattehöjningar (tax increases)
	...
1454	högkvalitativa (high quality)
1455	vackra (beautiful)
1456	klassiska (classic)
Normalised LIME score	
rank	**term**
1	vänsterregering (left-wing government)
2	fattigdomsbekämpning (poverty alleviation)
3	bidragsberoende (benefits dependency)
4	fridens (of peace)
5	arbetsföra (able to work)
	...
1454	som (as)
1455	ett (one)
1456	en (one)

Table 3: Results for the Moderates.

PCA ordering	
rank	**term**
1	budgetpropositionen (the budget bill)
2	arbetsmarknadspolitik (labor market policy)
3	samlingspartiet [Refers to the Moderates]
4	ungdomsarbetslösheten (youth unemployment)
5	skattesänkningar (tax cuts)
	...
1332	tillsammans (together)
1333	u (u)
1334	dam (lady)
Normalised LIME score	
rank	**term**
1	överläggningen (the deliberation)
2	moderatledda (moderate-led)
3	kd (abbrev. for Christian Democrat party)
4	skattesänkningarna (the tax cuts)
5	borgarna (the bourgeois [parties to the right])
	...
1332	har (have)
1333	av (of)
1334	för (for)

Table 4: Results for Social Democrats.

more to ensure that our results are *valid*. In other words, we want to ensure that our method measures what we intend to measure (Carmines and Zeller, 1979). In our case, this is whether a speech is representative of S or M.

Looking at how appropriate the terms are, as we did above, is a first step. This is also known as *face validity*, as we look if our method "appears to measure" what we want it to measure (Anastasi, 1976, pp. 139–140). Yet, face validity depends on many implicit decisions that vary between context and researcher. As such, we should look further if we wish to provide a more satisfactory validation. One good candidate for this is by looking at *construct validity* (Shadish et al., 2002; Carmines and Zeller, 1979). This refers to the degree to which we can use our results to say something about that what we aim to measure. One way to learn this here is to look at the wider context in which the terms the algorithm uses appear. For example, if a term used by the algorithm to assign a speech to S occurs in a context that defines S, this strengthens our case for construct validity. To see this, we can use keyword-in-context (KWIC), which looks at the *n* (here we choose 20) words before and after the term that interests us. In Table 5 we show this for one of the terms from the PCA analysis for S – *arbetsmarknadspolitik* (labour market policy). Here, we see that the context of the word indeed refers to policies close to S. In both cases, the term is used to call for more and new measures to regulate the labour market – something indicative of S. Similar examples for the words in Tables 3 – 4 are in the online appendix. As we have implemented KWIC in our algorithm, scholars can thus easily assess whether the same is true for any of the other terms and in this way better assess the validity.

"... enda åtgärd lösa detta, det behövs många åtgärder. Det handlar om ett gott företagarklimat, om en ny **arbetsmarknadspolitik**, om ytterligare utbildningssatsningar, om att bygga om — osv. med de förslag till åtgärder som vi ..." *"... single measure solve this, many measures are needed. It's about a good business climate, about a new **labour market policy**, about further training efforts, about rebuilding – etc. with the proposed measures that we ..."*
"... i arbete det finns individer som kommer att behöva säskilt stöd, och då behöver vi ha en bra **arbetsmarknadspolitik**. Men det är förstås inget egenvärde i att ungdomar som kan få jobb ändå ska vara i en ..." *"... in work there are individuals who will need separate support, and then we need to have a good **labour market policy**. But of course there is no intrinsic value in young people who can get a job still being in a..."*

Table 5: Keywords-in-context for the class-explanation feature *labour market policy* for the Social Democrats.

5.4 Explanations and Predictive Accuracy

Returning to individual instance explanations, we also wanted to investigate if the kind of words (domain specific or statistical distributions) occurring in an explanation have any relationship with the certainty of the model on those datapoints. We found domain specific words (here related to politics), along the positive PCA spectrum, while more common, general words had embeddings placing them towards the negative end. We find that data points where the explanation-words are predominantly positioned within the positive PCA spectrum (the sum of the PCA coordinates of the top-ten explanation features is positive) are cases where the model is more accurate. Compared to datapoints where explanations lie in the negative PCA space, there is an accuracy gain of roughly 10 percent (Table 6). Interestingly, this suggests that explanations containing domain specific, rarer words are correlated with the model's correctness, although the number of datapoints with domain specific explanations is quite small.

	Correct	**Incorrect**	**Acc**
Pos PCA sum	186	25	88.15
Neg PCA sum	1413	376	78.98

Table 6: Classifier performance on the validation set split based on the sum of PCA coordinates of the explanation provided by LIME.

6 Comparison to SP-LIME

Our method is comparable with SP-LIME, which aggregates individual LIME explanations. SP-LIME consists of three similar steps: post-hoc instance explanations extraction, sorting and exam-

Rank 1 SP-LIME example (true label S):
är (is), *det (the)*, *som (as)*, *den (the)*, *vi (we)*, *Natomedlemskap (NATO membership)*, *att (to)*, *du (you)*, *samlingsregeringen (the coalition government)*, **Vi (We)**
Rank 2 SP-LIME example (true label M):
frågorna (the questions), **protektionistiska (protectionist)**, **önskar (wish)**, **Det (The)**, **och (and)**, **Herr (Mr)**, *oerhört (incredibly)*, **handelsminister (Minister of Trade)**, **tackar (thanks)**, *de (the)*
...
Rank 12 SP-LIME example (true label M):
medelinkomsttagare (middle income earner), **avregleringar (deregulations)**, **vänster (left)**, **tvivelaktiga (questionable)**, *skattesänkningar (tax cuts)*, **Då (Then)**, **och (and)**, **Man (One/third person singular)**, *bostadsmarknaden (the housing market)*, *stöd (support)*
...
Rank 16 SP-LIME example (true label S):
borgarna (the bourgeois), *oss (us)*, *långtidsarbetslösa (long-term unemployed)*, *klyftorna (the cleavages)*, *det (the)*, *sjuka (sick)*, **rödgröna (red green)**[6], **Vi (We)**, **Låt (Let)**, **är (is)**

Table 7: Explanations provided by SP-LIME. Bold features indicate words contributing towards an M classification, while italic features do the same for S. Full results are in the online appendix.

ple extraction. In contrast to our proposed scoring functions, SP-LIME calculates the score for feature j as $I_j = \sqrt{\sum_{i=1}^{N} W_{ij}}$ where N is the number of data points in the explained dataset and W is the explanation matrix containing the local importance of the features. Based on this scoring, SP-LIME performs a greedy search to extract the top scoring data examples that also have the greatest coverage of distinct features. Therefore, the model explanation takes the form of a set number of text examples with their corresponding instance explanations, where the number of examples provided is defined by the user. Since the method performs a greedy search, the results are ordered by their contribution to how well they explain the model and how many unique features they cover.

We apply SP-LIME to the BERT classifier and extract the top 20 text examples that the explainability approach considers most representative. These contain 9 S examples and 11 M examples. A selected set of instance explanations can be seen in Table 7 and the full list is available in our online appendix. We can see the overemphasis of stop words especially in the top examples. Only a couple of the surfaced terms carry a political significance, and even those lack context and have debatable generalisability. Some of the examples provided by SP-LIME (see Top 12 and Top 16 in Table 7) are instances where human intuition is easier to align with. However SP-LIME in general does not provide a way to distinguish between the two types of contributing features that the current work targets. Finally, SP-LIME also differs from our method in the way it presents texts containing explanatory features. SP-LIME tries to find texts which have as many features as possible in one and the same text, while we choose to present many alternative contexts in which explaining feature words appear, motivated by social science use-cases.

7 Conclusion and Discussion

We have developed a new algorithm for extracting class explanations, which takes the distinction between stop words and content words into account. It thereby provides an alternative to prior methods like SP-LIME, which mixes explanations based on e.g. stop word frequency with the presence of certain domain-specific terms. Our motivation comes from the idea of human-grounded explainability: a useful explanation for a human will focus on content rather than stop words, while still being true to the model. In our case study, we demonstrated this for speeches from the Swedish parliament, with the task of explaining a binary classifier associating speeches to either of the two main parties. This is a difficult task, our human annotation experiment showed humans performing just better than random, potentially as they primarily looked for clues about policy. The machine learning models performed better, as they likely also managed to identify statistical speech patterns of speakers, which we saw in explanations where e.g. stop words inevitably appear. Our algorithm can not only identify these, but also separate them from explanations containing domain-specific words, hinting at policy, motivated by the needs of social scientists. Additionally, we find indications that domain-specific explanations cor-

relate with model performance. Patterns related to policy in our experiment may be more robust than learned speech patterns of stop words, which risks being influenced by single frequent individuals in the dataset, rather than capturing patterns common to a political party.

Future work will focus on systematic and extensive testing of the proposed methodology in order to evaluate it along the twelve properties proposed by Nauta et al. (2022). The focus should be on measuring the faithfulness to the underlying black box model, *correctness*, as well as a larger scale domain expert evaluation to measure how relevant and valid the explanations are (*context* and *coherence* properties). The generalisability will also be tested, by studying other domains and classification tasks.

Acknowledgments

This work was supported by the Wallenberg AI, Autonomous Systems and Software Program – Humanities and Society (WASP-HS) funded by the Marianne and Marcus Wallenberg Foundation and the Marcus and Amalia Wallenberg Foundation. RJ was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying Attention Flow in Transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197. ACL.

Anne Anastasi. 1976. *Psychological Testing*, 4 edition. Macmillan, New York, NY.

R. Arun, V. Suresh, and C. E. Veni Madhavan. 2009a. Stopword Graphs and Authorship Attribution in Text Corpora. In *2009 IEEE International Conference on Semantic Computing*, pages 192–196.

Rajkumar Arun, Ravi Saradha, V. Suresh, M. Murty, and C. Madhavan. 2009b. Stopwords and Stylometry: A Latent Dirichlet Allocation Approach. In *NIPS workshop on Applications for Topic Models*.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to Explain Individual Classification Decisions. *The Journal of Machine Learning Research*, 11:1803–1831.

Ulya Bayram, John Pestian, Daniel Santel, and Ali A. Minai. 2019. What's in a Word? Detecting Partisan Affiliation from Word Use in Congressional Speeches. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Edward Carmines and Richard Zeller. 1979. *Reliability and Validity Assessment*. Sage, Thousand Oaks, CA.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A Survey of the State of Explainable AI for Natural Language Processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459. ACL.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, MN. ACL.

Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning.

Kawin Ethayarajh and Dan Jurafsky. 2021. Attention Flows are Shapley Value Explanations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 49–54. ACL.

Lawrence Ezrow, Catherine de Vries, Marco Steenbergen, and Erica Edwards. 2011. Mean voter representation and partisan constituency representation: Do parties respond to the mean voter position or to their supporters? *Party Politics*, 17(3):275–301.

Simon Franzmann and André Kaiser. 2006. Locating Political Parties in Policy Space: A Reanalysis of Party Manifesto Data. *Party Politics*, 12(2):163–188.

Karin Friberg Heppin and Maria Toporowska Gronostaj. 2012. The Rocky Road towards a Swedish FrameNet - Creating SweFN. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 256–261. European Language Resources Association (ELRA).

John Hewitt and Percy Liang. 2019. Designing and Interpreting Probes with Control Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743. ACL.

John Hewitt and Christopher D. Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138. ACL.

Seth Jolly, Ryan Bakker, Liesbet Hooghe, Gary Marks, Jonathan Polk, Jan Rovny, Marco Steenbergen, and Milada Anna Vachudova. 2022. Chapel Hill Expert Survey trend file, 1999–2019. *Electoral Studies*, 75:102420.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and Understanding Neural Models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691. ACL.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-Hoc Interpretability for Neural NLP: A Survey. *ACM Computing Surveys*, 55(8):1–42.

Tim Miller. 2019. Explanation in Artificial Intelligence: Insights from the Social Sciences. *Artificial intelligence*, 267:1–38.

Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. 2022. From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI. *CoRR*, abs/2201.08164.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473. ACL.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101. ACL.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

William R. Shadish, Thomas D. Cook, and Donald T. Campbell. 2002. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin, Boston, MA.

Lloyd S. Shapley. 1952. *A Value for N-Person Games*. RAND Corporation, Santa Monica, CA.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. *CoRR*, abs/1905.06316.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5307–5315. ACL.

Constructing Pseudo-parallel Swedish Sentence Corpora for Automatic Text Simplification

Daniel Holmer, Evelina Rennes
Department of Computer and Information Science
Linköping University, Sweden
firstname.lastname@liu.se

Abstract

Automatic text simplification (ATS) describes the automatic transformation of a text from a complex form to a less complex form. Many modern ATS techniques need large parallel corpora of standard and simplified text, but such data does not exist for many languages. One way to overcome this issue is to create pseudo-parallel corpora by dividing existing corpora into standard and simple parts. In this work, we explore the creation of Swedish pseudo-parallel monolingual corpora by the application of different feature representation methods, sentence alignment algorithms, and indexing approaches, on a large monolingual corpus. The different corpora are used to fine-tune a sentence simplification system based on BART, which is evaluated with standard evaluation metrics for automatic text simplification.

1 Introduction

Automatic Text Simplification (ATS) is a sub-field of natural language processing mainly focusing on the automatic transformation a text from a complex form to a less complex form, and in that way make texts accessible for weaker readers. Even though the modern ATS techniques vary in scale and efficiency, there is one constant; the need for large parallel corpora of standard and simplified text, in order to train the simplification system.

The acquirement of such corpora is however not an easy task. One theoretical option is to collect manually created simplifications, but that process is incredibly time consuming and often not feasible due to the enormous amount of text that is required by modern ATS systems.

A second option is to leverage already existing sources of parallel texts. One common example is the collection of articles from Wikipedia alongside their Simple Wikipedia counterpart. However, Xu et al. (2015) identified numerous problems to the dual Wikipedia approach, for example the fact the simple article most often is not a rewrite of the standard article. This can lead to a variation of the content in the articles that is large enough to make them unsuitable to be included in an aligned corpus. Moreover, the Simple English Wikipedia presents a limitation in text simplification research due to its sole availability in the English language. One way to overcome this problem is to translate the English texts into another language. For instance, Sakhovskiy et al. (2021) translated the WikiLarge dataset (Zhang and Lapata, 2017) into Russian.

Another possibility would be to follow the approach of Kajiwara and Komachi (2018), where a monolingual sentence corpus is divided into a standard and simplified part, and aligned with the best sentence matches between the two corpora. The result is a "pseudo-parallel" monolingual corpus; a parallel monolingual corpus that has been aligned with an unsupervised alignment algorithm rather than been manually constructed or collected from an already divided source, circumventing the previously mentioned problems. The approach was proven to perform well for both English and Japanese domains.

The aim of the work presented in this paper was two-fold. First, we aimed to create Swedish pseudo-parallel sentence simplification corpora[1] from a single monolingual Swedish sentence corpus. Second, we aimed to investigate how different methods and techniques used during the creation influence the performance of sentence simplification systems trained on the different cor-

[1]The corpora are made available at: https://github.com/holmad/Constructing-Pseudo-parallel-Swedish-Sentence-Corpora-for-Automatic-Text-Simplification

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 113–123

pora. The research question we explored was:

- For different alignment and embedding techniques, which alignment thresholds produce corpora that when used to fine-tune a BART model, produce sentence simplifications with the highest BLEU and SARI scores?

2 Related work

Data-driven approaches are common for most modern research in sentence simplification (Alva-Manchego et al., 2020). Data-driven does—in this context—refer to the collection of parallel corpora of standard-simple sentence pairs. These corpora are then used to train simplification systems by considering the simplification task as monolingual machine translation.

Much research has been conducted by exploiting the standard and simple versions of the English Wikipedia (Zhu et al., 2010; Coster and Kauchak, 2011; Woodsend and Lapata, 2011; Hwang et al., 2015; Zhang and Lapata, 2017). Additionally, the Newsela corpus (Xu et al., 2015) has been used for the creation of aligned corpora (Alva-Manchego et al., 2017; Zhao et al., 2018) , much alike Wikipedia. The Newsela corpus contains $1,130$ standard news articles, combined with up to five simplifications for each given article. The simplifications are created by professional writers, which overall should be an improvement in quality over the simplifications in the Simple English Wikipedia, which are produced by volunteers (Alva-Manchego et al., 2020). In a Swedish context, Rennes (2020) compiled a corpus of $15,433$ unique sentence pairs derived from the websites of Swedish authorities and municipalities. This comparatively small resource is the only available aligned corpus of standard-simple sentence pairs for Swedish.

In contrast to the previously mentioned corpora, which are based on alignment of sentences that are extracted from one source of standard sentences and another source of simplified sentences, the construction of a pseudo-parallel monolingual corpus includes the process of deciding if every given sentence should be considered as one of standard or less complexity. For this task, Kajiwara and Komachi (2018) calculated the, for English text widely used, Flesch Reading Ease Score (FRES) for each sentence, and in that way determined its complexity. The Swedish counterpart to FRES is called *Läsbarhetsindex* (LIX) (Björnsson, 1968). Since LIX only measures the lengths of words, sentences, and ratios of long words, additional text complexity metrics have been developed for Swedish texts, such as the SCREAM (Falkenjack et al., 2013; Falkenjack, 2018) and SVIT (Heimann Mühlenbock, 2013) measures.

With MUSS, Martin et al. (2022) implemented a method to align paraphrases based on their similarity measures. In order to train a simplifier to produce simplifications, as opposed to just paraphrases, the authors employed ACCESS (Martin et al., 2020). ACCESS enables controllable output of sequence-to-sequence models by including special control tokens, that—among other things—can be used to limit the length of decoder output.

2.1 BART

BART (Lewis et al., 2020) is an autoencoder for pretraining models for sequence-to-sequence tasks. A BART model is trained by inputting text corrupted with a noising function, and learning to reconstruct the text to its original state. Hence, it is a *denoising* autoencoder. BART utilises a bidirectional encoder[2], where random tokens are masked and the document is encoded by considering tokens in both directions. For the prediction of the masked tokens, each token is predicted independently by considering the entire input sequence. Since text-generation is a task that only considers the current and previous input, a standard BERT model is unsuitable for text generation[3] (Lewis et al., 2020). With BART, the bidirectional encoder is paired with an auto-regressive left-to-right decoder. The auto-regressive decoder predicts tokens by considering the current token combined with the leftward context, and can therefore generate new text.

The combination of the two components allows BART to apply any noising function, compared to previous autoencoders that are tailored for a specific function (Lewis et al., 2020). The number of possible pre-training tasks that can be employed by BART is therefore also significantly larger than, for example, BERT.

[2]The structure is very similar to that of BERT (Devlin et al., 2019), but some discrepancies can be noted. For instance, BART replaces ReLU with GeLU activation functions. See Lewis et al. (2020) for details.

[3]However, the weights of a BERT model can be used in a warm-start procedure of an encoder-decoder model to achieve similar capabilities. See for example Rothe et al. (2020) and Monsen and Jönsson (2021)

3 Data

We used several different datasets for different tasks. Table 1 provides an overview of the datasets used.

The Stockholm-Umeå Corpus (or *SUC*) (Gustafson-Capková and Hartmann, 2006) is a balanced corpus of Swedish texts from the 1990s. The style of text is varied, and it is sometimes used as a baseline for standard use of the Swedish language during the time period (see for example Pettersson and Nivre (2011)). In total, the corpus consists of $1,166,593$ tokens and $74,245$ sentences.

The NyponVilja dataset consists of OCR scans of books from Sweden's largest publisher of easy-to-read books, *Nypon och Vilja Förlag*, targeting children and youths. Each book is graded by human experts with a readability level, where level 1 denotes a book that is the easiest to read and level 6 denotes books that provide the most challenge for the readers.

The CCNET dataset is provided by Common Crawl[4], a non-profit organisation that uses web crawlers to collect an enormous amount of text data from all around the web, and makes it freely available to the public. The organisation collects and publishes a new data snapshot approximately 10 times a year[5], each snapshot in the size range of $\approx$ 100–300TB whereof 20–30 TB is raw text data.

We used the Swedish part of the CC-100 dataset, previously used to recreate the training of XLM-R (Conneau et al., 2020), for the sentence alignment task. The dataset was created by researchers at StatMT[6], by applying the CC-Net pipeline to extract datasets for 100 different languages from the Common Crawl snapshots created during the time period January–December 2018. The Swedish dataset comprises 80GB uncompressed text, in the form of $580,387,314$ paragraphs. From these paragraphs, $61,959,899$ sentences were extracted for further pre-processing and annotation.

The data was further prepared for alignment by roughly following the procedure in Raffel et al. (2020). However, an additional step was introduced to rearrange the data from paragraphs to sentences. This step was added since the task is to align sentences, not paragraphs. It was therefore also necessary to annotate the dataset on the sentence level.

We used the SAPIS (Fahlborg and Rennes, 2016) pipeline to tokenise each sentence with Efselab (Östling, 2018), and to annotate each sentence with a subset of the SCREAM metrics previously identified by Santini et al. (2020).

PK18 (Lindberg and Kindberg, 2018) is a corpus totalling 1,005 texts pairs. Each pair consist of an original version of the given text, and a simplified version of the same text. The texts origin from four Swedish organisations and municipal-, regional- and state departments; *Riksförbundet för utvecklingsstörda barn, ungdomar och vuxna (FUB)*, *Linköpings Kommun*, *Region Östergötland*, and *Specialpedagogiska myndigheten (SPSM)*. The simplified versions were written by experts, and were manually aligned with the corresponding original version of the texts.

PK18 is currently the largest available corpus suitable for use as a gold standard for the evaluation of Swedish ATS systems. Since this work focused on sentence-level simplification, only the pairs aligned in a 1-1 manner were used. The result was a dataset of 467 sentence pairs, with the purpose of being used as the test dataset for the fine-tuned text simplification system.

4 Implementation

This section describes the creation of the pseudo-parallel corpora and their usage in text simplification systems. The procedure can be outlined in four steps. First, the sentences were classified as being of either standard or simple complexity. Second, the sentences were aligned. Third, the different corpora were provided as training data to fine-tune multiple text simplification systems. Finally, the performance of each of the systems was assessed with standard evaluation metrics.

4.1 Labelling of sentences as standard or easy

Following Kajiwara and Komachi (2018), the sentence dataset was divided into two subsets, one with standard sentences and one with easy sentences.

We used a classification model to determine if the sentences from the CCNet dataset should be seen as "standard" or "easy". The model was realised with the implementation of Support Vec-

[4] https://commoncrawl.org

[5] Each snapshot can be found at https://index.commoncrawl.org

[6] https://data.statmt.org/cc-100/

Dataset name	Sentences	Tokens	Usage
SUC	74,243	1,166,593	Standard sentences used for training of the SVM sentence classifier.
NyponVilja	54,938	459,540	Easy sentences used for the training of the SVM sentence classifier.
CCNet subset	61,959,899	832,996 921	Sentences which were classified as either easy or standard, and then aligned to form the easy/standard sentence pairs of the pseudo-parallel corpora.
PK18 subset	467 (sentence pairs)	7,873 (standard) 6,429 (simplified)	A manually annotated dataset that is used for evaluation of the sentence simplifier trained on the aligned corpora.

Table 1: Overview of the different datasets used.

tor Machine (SVM) found in the Python library scikit-learn (Pedregosa et al., 2011). We annotated each sentence with a subset (described in Santini et al. (2020)) of the text complexity metrics from SCREAM (Falkenjack et al., 2013; Falkenjack, 2018), previously known to predict text complexity in Swedish. Since the metrics vary in scale (for instance, some metrics are ratios while other are raw frequencies), they were standardised by removing the mean and scale to unit variance, before being used as features to represent a sentence in the SVM.

The SVM was then trained with the standard sentences (from SUC) and the easy sentences (from NyponVilja) as class labels. A 10-fold cross-validation process was applied to evaluate the model performance. Averaged over all folds, the SVM classifier performed with an F1-score of 82%. This SVM classifer was then used to assign all sentences from the CCNet dataset as of either standard or easy complexity.

4.2 Alignment of sentences

The alignment of sentences labelled in the previous section can be divided into two categories: alignments based on similarities of individual word embeddings between sentences, and alignment based on the similarity of embeddings of whole sentences.

A common functionality between the two approaches is the ability to filter the resulting corpus with regard to the alignment threshold. A higher threshold would allow fewer sentence pairs to be included in the corpus, but the pairs that were included would be more similar according to the cosine distance, and therefore probably of higher quality. Inversely, a lower threshold would include more sentence pairs, but their similarity would on average be lower. To investigate this trade-off, corpora with the alignment threshold of both 0.8 and 0.9 were created.

4.2.1 Word-based embeddings

At its core, this approach is based on the method originally proposed by Song and Roth (2015) and later used by both Kajiwara and Komachi (2018) and Rennes (2020), where sentences were aligned according to their similarity at the word level. Different alignment algorithms were used to perform the task, where Kajiwara and Komachi (2018) implemented *Average (AA)*, *Maximum (MA)*, and *Hungarian (HA)* alignment algorithms, as well

as the *Word Mover's Distance (WMD)*. Rennes (2020) implemented the *AA*, *MA*, and *HA* alignment algorithms.

The main difference in this work when compared to the aforementioned works is the much increased dataset size; an increase of several million sentences. This brings forth some additional challenges, mainly regarding the computational complexity during the alignment process. For this reason, we only used the *AA* and *MA* algorithms. Both *HA* and *WMD* resulted in a dramatic increase in the required computations, which were not feasible to perform given the available hardware and time frame.

Average alignment similarity (AAS) calculates the pairwise cosine similarities between all the words of sentence x and sentence y and averages them over the number of pairs (see Equation 1).

$$AAS(x, y) = \frac{1}{|x||y|} \sum_{i=1}^{|x|} \sum_{j=1}^{|y|} cos(x_i, y_j) \quad (1)$$

Maximum alignment similarity (MAS) can be seen as a refinement of the AAS, since it does only take into account the best (maximum cosine similarity) word pair between sentence x and sentence y (see Equation 2).

$$MAS_{asym}(x, y) = \frac{1}{|x|} \sum_{i=1}^{|x|} \underset{j}{max}\, cos(x_i, y_j) \quad (2)$$

Equation 2 describes an *asymmetric* similarity, meaning that there will be different total similarity scores depending on if each of the words of sentence x gets paired with its maximum similarity in sentence y, and vice versa[7]. Therefore, to get a symmetric MAS, we add the averages of the asymmetric MAS(x, y) and MAS(y, x), as described in Equation 3.

$$MAS(x, y) = \tfrac{1}{2} MAS_{asym}(x, y) + \tfrac{1}{2} MAS_{asym}(y, x) \quad (3)$$

In earlier works, MAS has shown to be well performing, and the alignment algorithm of choice of both Kajiwara and Komachi (2018) and Rennes (2020).

[7]Unless the sentences are identical, but that would of course make the whole alignment procedure unnecessary

Another consequence of the increased amount of data is the need to restrict the search problem during the alignment process. Even though only the computationally least demanding alignment algorithms were used, to calculate the cosine distances in a *N:M* manner (that is, between every easy sentence and every standard sentence) would be too computationally expensive. Therefore, a more efficient method of calculating the similarities was implemented.

We used MinHashLSH[8] to construct an index from the easy sentences, and query the index with the standard sentences to create a mapping of potential sentence pairs for alignment (see step 1, 2, and 3 in Figure 1). MinHash allows the matching of sentences that share fewer features in the syntactic sense, than for example SimHash as proposed in earlier works, but still set a requirement that the sentences have to share similarities at a given threshold. For this work we used the Jaccard similarity of `0.5` for a sentence pair to be considered a possible match. This allowed for a relatively large range of possible matches, but still dramatically reduced the search space. The index was constructed with a feature window of `5` and the `num_perm` parameter of `16`.

After the construction of the index and the extraction of possible matching sentences, we used Fasttext[9] pre-trained Swedish word vectors to embed every word in every sentence of the matching pairs. In order to reduce the memory footprint of the vectors, we reduced the dimensions from the default `300` to `100` dimensions. This allowed for more vectors to be loaded in memory, and allowed larger batches of computations of several sentences at once. This significantly improved the computational overhead for the alignment module. The embeddings of the words in the matched sentences then got passed to the alignment module (see steps 4, 5, and 6 in Figure 1).

4.2.2 Sentence-based embeddings

For this approach, each sentence was represented as a sentence embedding via Swedish sentence-BERT (Rekathati, 2021). Each embedding from the standard bucket was compared to all of the embeddings from the easy bucket, and the pair of standard and easy sentences with the highest

[8]from the datasketch package `http://ekzhu.com/datasketch/lsh.html`

[9]`https://fasttext.cc/docs/en/crawl-vectors.html`

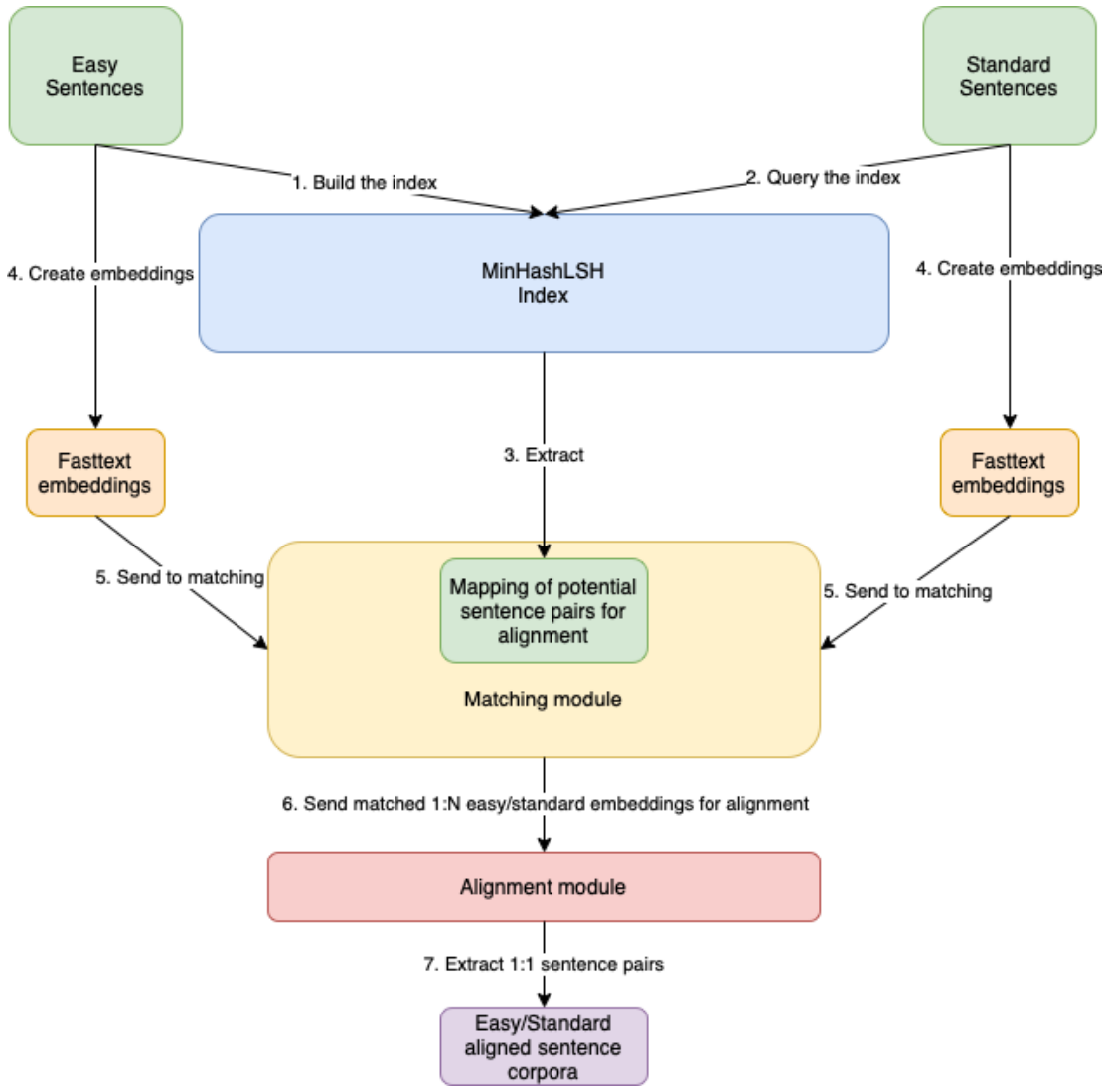


Figure 1: High-level overview of the alignment of sentences with the word-level Fasttext embeddings.

cosine similarity was aligned and added to the corpus. To speed up this process and forgo the quadratic complexity of an exhaustive search, the embeddings of the easy sentences were indexed using Faiss (Johnson et al., 2019). Since Faiss requires all embeddings to be loaded into memory when constructing the index, we employed PCA to reduce the output dimension of the sentence transformer model from 768 to 128[10]. The slight reduction in quality for each embedding was deemed to be outweighed by the ability to use all easy sentence embeddings for the index training and construction.

For this work, we used the `IVFFPQ`-index from Faiss, which utilises both coarse- and fine quantisation to reduce both search times and index disk size. The index was trained with the parameters `nlist=2048` (the number of Voronoi cells), `nbits=8` (the number of bits to represent the codes per each subvector), and `M=8` (the number of subvectors per vector). Additionally, each embedding vector was normalised to support measuring cosine distances as opposed to Euclidean distances.

For the standard sentences, each sentence embedding was queried to the index, and the easy sentence with the highest cosine similarity to the queried standard sentence was extracted if it adhered to the given similarity threshold set for the current corpus.

[10]This process was based on the following code from SentenceTransformers https://github.com/UKPLab/sentence-transformers/blob/master/examples/training/distillation/dimensionality_reduction.py

Embedding type	Word alignment algorithm	Threshold	Sentence pairs	Avg. sentence length (easy)	Avg. sentence length (standard)	BLEU	SARI
-	-	baseline	-	-	-	22.81	12.80
word	AA	0.8	440,259	8.24	12.76	10.17	33.11
word	AA	0.9	40,014	7.16	8.05	17.29	28.31
word	MA	0.8	442,152	8.25	12.77	9.53	**33.24**
word	MA	0.9	40,017	7.16	8.05	16.71	29.51
sentence	-	0.8	6,560,372	7.09	12.25	4.04	30.43
sentence	-	0.9	652,964	6.23	9.20	3.64	30.29

Table 2: The created corpora and their evaluation scores when used to train the simplification system.

4.3 Simplification module

Each corpus was used to fine-tune a simplifier based on a Swedish BART model[11], developed by KBLab. They pre-trained the model on approximately 80GB of text (around 15B tokens) with the help of Fairseq[12], and subsequently converted it to be compatible with the Huggingface Transformers Python-library (Wolf et al., 2020). The pre-trained model consisted of approximately 139M parameters.

In our work, the fine-tuning and evaluation pipeline was in large part built with the Transformers library. Each sentence pair were tokenised using the pre-trained model's tokeniser with the `AutoTokenizer` class and the model was loaded using the `AutoModelForSeq2SeqLM` class. For the fine-tuning, the hyperparameters were consistent for all models, with the `learning rate=3e-05` and `batch size=32`. Furthermore, the number of `warmup steps` were `500` and the `weight decay=0.1`. The optimisation algorithm was the default `AdamW` and each simplification model was fine-tuned for between `1` and `10` epochs, depending on corpus size. In general, the hyperparameters were kept close to the default values, and the ones we experimented with only showed minor differences in performance.

From each corpus, 90% of the sentence pairs were used as training data, and 10% were used as validation data.

[11] https://huggingface.co/KBLab/bart-base-swedish-cased

[12] https://github.com/facebookresearch/fairseq

4.4 Evaluation

For the evaluation, we applied two metrics commonly used for the assessment of ATS systems – BLEU and SARI. BLEU (BiLingual Evaluation Understudy) (Papineni et al., 2002) is calculated with modified unigram precision and a brevity penalty factor between a target and reference sentence. The SARI metric (Xu et al., 2016) compare **s**ystem output **a**gainst **r**eferences and against the **i**nput sentence. The purpose of SARI is to quantify the simplification of sentences based on words that are *added*, *deleted*, or *kept* by the simplification system. (Alva-Manchego et al., 2020) describes the intuition behind SARI as that the system is rewarded for the *addition* of n-grams that occur in any of the references but not in the input, the *keeping* of n-grams both in the output and the references, and the avoidance of over-*deleting* n-grams.

Unfortunately the PK18 subset is limited by its small size, but it is to the best of the authors knowledge the only manually aligned simplification dataset in Swedish, and future studies would benefit from a larger, high quality dataset. For this study, we did however use the PK18 subset of 467 manually aligned sentence pairs to evaluate the performance of the BART simplifiers trained on the different generated corpora. Each sentence pair was passed as test data, and BLEU and SARI metrics were calculated. As a baseline, we calculated the BLEU and SARI metrics for the test dataset when no simplification was performed (i.e. the original sentence was used as the system output sentence and the gold standard simplified sentence was used as the reference sentence). For both BLEU and SARI calculations,

we used the implementation from EASSE (Alva-Manchego et al., 2019).

5 Results

For simplicity, the created corpora are referred to with the notation of *[embedding type]_[word alignment algorithm]_[alignment threshold]*. For example, the corpus in the second row of Table 2 is referred to as `word_AA_0.8`.

In Table 2, the results of the corpora created with the alignment and embedding methods described in Sections 4.2.1 and 4.2.2 are presented. The baselines for BLEU and SARI were calculated as described in Section 4.4 (i.e. they were calculated as if no simplification was conducted at all).

All of the corpora performed better than the baseline SARI. However, the best performance was shown by both word embedding-based corpora with a filtering threshold of 0.8, with a SARI score of over 33. This is higher than the corpus aligned with the help of sentence embeddings, which had a SARI of 30.43. Of the two best-performing word embedding-based corpora, the one aligned with the MA algorithm performed with a slightly higher SARI score than the AA one.

For the BLEU score all of the corpora showed lower values than the baseline. The corpus based on word embeddings and with an alignment threshold at 0.9 did however show BLEU scores fairly close to the baseline. The rest of the corpora performed significantly lower.

It is clear that the number of sentence pairs is closely related to the alignment threshold. For all embedding type/word alignment algorithm combinations, the corpus with a higher threshold also consisted of fewer sentence pairs than their lower threshold counterparts.

6 Discussion

In this section the results for the different conducted experiments will be discussed.

6.1 Alignment results

Inspecting the results in Table 2, a first thing to note is that all of the models fine-tuned on the corpora performed with higher SARI scores than the baseline. Furthermore, the two corpora created using embeddings on the word level and the sentence alignment threshold of 0.8, `word_AA_0.8` and `word_MA_0.8`, showed the highest SARI scores (33.11 & 33.24) in this study.

On the other hand, the `word_AA_0.9` and `word_MA_0.9` corpora showed significantly higher BLEU scores than the rest, while at the same time exhibiting relatively low SARI scores. One explanation for this behaviour is that the simplifications from the models fine-tuned on these corpora often include only minor changes to the original sentence. In some cases, no change from the original sentence can be observed at all. As a consequence, since few (or none) *add*, *delete*, or *keep* operations can be rewarded, the SARI score will be kept low. Inversely, the similarity between the original and output sentences will benefit the BLEU score. The evaluation dataset contains, in many cases, small differences between the standard and simplified sentence, with only small parts of information either added or deleted. This in turn leads to a situation where the reference and original sentences are so similar that a (relative to the baseline) high BLEU can be achieved by just keeping the original sentence.

When looking at both the corpora based on sentence embeddings (`sentence_0.8` and `sentence_0.9`), it can be noted that the SARI scores are somewhat average compared to the other corpora. The BLEU scores are however significantly lower. One possible explanation for this behaviour could be that BLEU is more restrictive than SARI, in the sense that the same n-gram have to be present in both the target and reference sentence for BLEU. Since the sentence embeddings are a semantic representation of the sentence, two sentences could have high similarity scores on the sentence level while having a low ratio of shared n-grams.

Overall, the `word_MA_0.8` corpus performed with the most balance between the BLEU and SARI scores, closely followed by `word_AA_0.8`.

6.2 Evaluation metrics

While BLEU has been used as a metric for the evaluation of automatic text simplification systems, it is problematic to use. Sulem et al. (2018) showed how BLEU fails to serve as a useful evaluation metric for sentence splitting operations. Since the corpora created in this work are aligned in a sentence-to-sentence manner, this point is of less importance for this specific evaluation. However, the authors did also find that BLEU of-

ten *negatively* correlates with simplicity, and may penalise simpler sentences instead of rewarding them. To rely on BLEU as the only metric for evaluation is therefore not to recommend. In this work, its main purpose is instead to indicate the *similarity* of the reference and system output, not necessarily the difference in *simplicity*. For example, the BLEU metric gives support to the observations that the simplified sentences of the models fine-tuned on `word_AA_0.9` and `word_MA_0.9` in many instances is just a cut-off version of the standard sentence, where either the beginning or end of the sentence have been removed. For this particular behaviour, the BLEU metric provided valuable information despite its other apparent flaws in the task of text simplification.

Another thing to note is the low BLEU scores overall, but in particular for the corpora based on sentence embeddings. The low overall scores can probably, as also observed by Kajiwara and Komachi (2018), partly be attributed to the lack of multiple reference sentences in the test dataset. An additional contributing factor to low scores for the corpora based on sentence embeddings is probably the behaviour that sentences with named entities often get aligned with sentences containing completely different entities. This leads to a corpus of sentences with a lower ratio of exact word-to-word matches. When evaluating simplification models fine-tuned on these corpora, the BLEU metric would probably be more affected by this than the SARI metric.

In recent years, much of the published research on text simplification systems has used SARI as an evaluation metric. One of its main merits is that it is good at assessing a system's ability to perform lexical paraphrasing. (Alva-Manchego et al., 2021) suggest using a combination of multiple metrics to capture different aspects of text simplification. In future studies it would be interesting to implement a wider array of metrics, for example BERTScore (Zhang et al., 2020) or METEOR (Denkowski and Lavie, 2011), to further examine the quality of the corpora.

7 Conclusion

The aim of the work presented in this paper was to create a Swedish pseudo-parallel sentence simplification corpus from a single monolingual Swedish sentence corpus, and to investigate how different methods and techniques used during the creation influence the performance of sentence simplification systems trained on the different corpora.

From the results, it can be seen that the model fine-tuned on a corpus created with word-based embeddings, the Maximum Alignment algorithm, and an alignment threshold of 0.8 performed with the best SARI and acceptable BLEU scores. It is however unclear how much the different indexing methods impacted the performance of the alignment process, and exactly how the quality of the corpora was affected.

Both the investigated methods of creating pseudo-parallel corpora for sentence simplification show promising results. Future studies should conduct a further investigation on different parameter choices, mainly when constructing the indices to help the alignment process, and explore how they impact the quality of the corpora. The resulting corpora should also be thoroughly evaluated with regard to different aspects of text simplification; with a combination of qualitative evaluations, additional evaluation metrics, and a larger test dataset.

References

Fernando Alva-Manchego, Joachim Bingel, Gustavo Henrique Paetzold, Carolina Scarton, and Lucia Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 295—-305, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. EASSE: Easier automatic sentence simplification evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 49–54, Hong Kong, China. Association for Computational Linguistics.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, pages 1–87.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2021. The (Un)Suitability of Automatic Evaluation Metrics for Text Simplification. *Computational Linguistics*, 47(4):861–889.

Carl-Hugo Björnsson. 1968. *Läsbarhet*. Liber, Stockholm, Sweden.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

William Coster and David Kauchak. 2011. Simple English Wikipedia: A New Text Simplification Task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the sixth workshop on statistical machine translation*, pages 85–91.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Daniel Fahlborg and Evelina Rennes. 2016. Introducing SAPIS - an API service for text analysis and simplification. In *The second national Swe-Clarin workshop: Research collaborations for the digital age, Umeå, Sweden*.

Johan Falkenjack. 2018. *Towards a model of general text complexity for Swedish*. Licentiate thesis, Linköping University Electronic Press.

Johan Falkenjack, Katarina Heimann Mühlenbock, and Arne Jönsson. 2013. Features Indicating Readability in Swedish Text. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NoDaLiDa-2013), Oslo, Norway*, number 085 in NEALT Proceedings Series 16, pages 27–40. Linköping University Electronic Press.

Sofia Gustafson-Capková and Britt Hartmann. 2006. Manual of the stockholm umeå corpus version 2.0. Technical report, Stockholm University.

Katarina Heimann Mühlenbock. 2013. *I see what you mean. Assessing readability for specific target groups*. Dissertation, Språkbanken, Dept of Swedish, University of Gothenburg.

William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. Aligning sentences from Standard Wikipedia to Simple Wikipedia. In *HLT-NAACL*, pages 211–217.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Tomoyuki Kajiwara and Mamoru Komachi. 2018. Text simplification without simplified corpora. *The Journal of Natural Language Processing*, 25:223–249.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Maja Lindberg and Erik Kindberg. 2018. Proffskorpus - korpusdokumentation. Internal report.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2022. Muss: Multilingual unsupervised sentence simplification by mining paraphrases. In *Proceedings of the Language Resources and Evaluation Conference*, pages 1651–1664, Marseille, France. European Language Resources Association.

Julius Monsen and Arne Jönsson. 2021. A method for building non-english corpora for abstractive text summarization. In *Proceedings of CLARIN Annual Conference 2021*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct):2825–2830.

Eva Pettersson and Joakim Nivre. 2011. Automatic verb extraction from historical swedish texts. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 87–95.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Faton Rekathati. 2021. The KBLab blog: Introducing a Swedish sentence transformer. https://kb-labb.github.io/posts/2021-08-23-a-swedish-sentence-transformer/.

Evelina Rennes. 2020. Is it simpler? an evaluation of an aligned corpus of standard-simple sentences. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI@LREC), Marseille, France.*, pages 6–13.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks.

Andrey Sakhovskiy, Alexandra Izhevskaya, Alena Pestova, Elena Tutubalina, Valentin Malykh, Ivan Smurov, and Ekaterina Artemova. 2021. Rusimplesenteval-2021 shared task: evaluating sentence simplification for russian. In *Proceedings of the International Conference "Dialogue"*, pages 607–617.

Marina Santini, Arne Jönsson, and Evelina Rennes. 2020. Visualizing facets of text complexity across registers. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI@LREC). Marseille, France*, pages 49–56.

Yangqiu Song and Dan Roth. 2015. Unsupervised sparse vector densification for short text similarity. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1275–1280.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Bleu is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 738–744, Brussels, Belgium.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173, Brussels, Belgium. Association for Computational Linguistics.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A Monolingual Tree-based Translation Model for Sentence Simplification. *23rd International Conference on Computational Linguistics*, (August):1353–1361.

Robert Östling. 2018. Part of speech tagging: Shallow or deep learning? *North European Journal of Language Technology*, 5:1–15.

Who said what? Speaker Identification from Anonymous Minutes of Meetings

Daniel Holmer, Lars Ahrenberg, Julius Monsen, Arne Jönsson
Department of Computer and Information Science
Linköping University, Sweden
daniel.holmer|lars.ahrenberg|julius.monsen|arne.jonsson@liu.se

Mikael Apel
Sveriges Riksbank
mikael.apel@riksbank.se

Marianna Blix Grimaldi
The Swedish National Debt Office
marianna.blixgrimaldi@riksgalden.se

Abstract

We study the performance of machine learning techniques to the problem of identifying speakers at meetings from anonymous minutes issued afterwards. The data comes from board meetings of Sveriges Riksbank (Sweden's Central Bank). The data is split in two ways, one where each reported contribution to the discussion is treated as a data point, and another where all contributions from a single speaker have been aggregated. Using interpretable models we find that lexical features and topic models generated from speeches held by the board members outside of board meetings are good predictors of speaker identity. Combining topic models with other features gives prediction accuracies close to 80% on aggregated data, though there is still a sizeable gap in performance compared to a not easily interpreted BERT-based transformer model that we offer as a benchmark.

1 Introduction

Attributing a text or a part thereof to an agent is a well-established sub-field of computational linguistics. Apart from the traditional task of author attribution, it has also been applied in social media studies, to the identification of speakers in fiction dialogues, and for detection of plagiarism. In this work, we study a new but related problem: identifying speakers at meetings from anonymous minutes issued afterwards.

The data at hand are minutes, in Swedish, from the monetary policy meetings of the Riksbank's Executive board. The main monetary policy objective is to keep inflation low and stable, close to the target of 2 percent. The key issue at the meetings is to decide on the policy rate, and, since the global financial crisis in 2007-2009, on purchases of financial assets. Minutes from meetings like these are not only common for central banks but also, for instance, corporates, c.f. (Agarwala et al., 2022; Schwartz-Ziv and Weisbach, 2013).

Until June 2007 the minutes of the Swedish Riksbank's monetary policy meetings gave an anonymised account of the deliberations. Since then, however, the identity of a board member is revealed in the minutes so that it is possible to know which member expressed which opinion during the meeting. This change towards increased transparency is of great interest to researchers on economic policy-making and there is a growing literature in this area (Hansen et al., 2018). It could potentially affect board members' incentives and behaviour in different ways, not least because the minutes are published only around two weeks after a meeting.

Following the theoretical literature increased transparency can have different effects. It can make agents prepare more thoroughly – a disciplinary effect (Holmström, 1999). It can also make agents behave differently due to career concerns, either by making them less inclined to oppose to the majority view – a herding, or conformism, mechanism – or by making them instead want to distinguish themselves more from others – an anti-herding or exaggeration mechanism. It may also make agents more committed to stick to a specific opinion once they have expressed it and less willing to change their mind, even if circumstances change (Falk and Zimmermann, 2018).

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 124–134

Swedish	English translation
Vice riksbankschef A inledde diskussionen med att uttrycka sitt stöd för det B sade om behovet av att ha en bredare ansats när man analyserar skälen till den låga inflationen. Här är det, menade han, viktigt att ta hänsyn till både efterfråge- och utbudsfaktorer. Att fokus varit ensidigt kan möjligen vara förståeligt i länder som befunnit sig i krisens epicentrum, fortsatte A. Där har stora negativa effekter på produktion, sysselsättning och arbetslöshet helt dominerat både debatten och den ekonomiska politikens inriktning.	Deputy Governor A started the discussion by expressing his support for what B had said about the need for a broader approach when analysing the reasons for the low inflation rate Here, he said, it is important to consider factors of both demand and supply. That there has been a one-sided focus may be understandable in countries that have been at the epicentre of the crisis, A continued. There great negative effects on production and employment have dominated both the debate and the direction of economic policies.

Table 1: Extract from a contribution.

Here we are not concerned with transparency effects as such, rather we want to find out what features and methods would enable us to trace the behaviour of individual members when conditions are changed, from a state where views, but not identities, are reported in the minutes, to a state where both identities and views are revealed. The study can be seen as a first contribution to the development of automatic tools that can support transparency studies by analysing minutes of meetings created under different conditions.

In this study we investigate the problem of predicting agent identities under a supervised condition, using minutes from the period September 2007 to April 2018 for experiments. During this period the board has had six members at any given time, but as members have limited periods of service, altogether twelve people have served on the board. We are looking for features of the board members that can be assumed to be relatively stable over time, and so be used for identification. The study is thus an experiment in de-anonymisation, which has been defined as a reverse engineering process in which de-identified data are cross-referenced with other data sources to re-identify the personally identifiable information[1]. The data to be re-identified are participants' contributions to the discussions preceding the vote on policy rate as they are reported in the minutes. The primary data used for cross-referencing are speeches made by the members to private and public audiences outside of board meetings. Both the minutes of the meetings and the speeches are publicly available on the Riksbank's website.

The minutes are compiled by a secretary who has access to recordings of the meeting. Discussions and decisions are reported in detail using a formal writing style where sentences are well-formed and punctuation formal. For an example, see Table 1. During a meeting a member may make several contributions and the start of a new contribution is usually marked in the minutes by a reference including the title and full name of the member. A contribution can be short, only a few sentences, but sometimes as long as several paragraphs. The minutes may sometimes partly be based on written notes provided by members but we do not know to what extent this happens nor how much editing is done.

The aims of the study are three-fold: 1) to compare the performance of several machine learning methods on this task, all of which have been successfully applied to attribution tasks in the past; 2) to identify features of members and their contributions that can aid de-anonymisation; 3) to establish a benchmark for what can likely be achieved on anonymised minutes under an unsupervised condition.

The methods investigated are:

- Burrows' Delta (Burrows, 2002)
- A Support Vector Machine (SVM)
- A Multi-layer Perceptron (MLP)
- Two ensemble methods of SVMs and MLPs
- A Swedish BERT model (Malmsten et al., 2020) fine-tuned for sequence classification

The paper is organised as follows. In section 2 we report related work. In section 3 we describe our data and the preprocessing we have applied. In

[1] https://codata.org/rdm-glossary/de-anonymization/

section 4 we describe the features we have used in the study and in particular the topic model we have used. Section 5 reports our experiments and the final sections discuss our results and report our conclusions.

2 Related work

We have not been able to find studies that perform speaker attribution under equal circumstances. A study on cabinet meetings (Ruppenhofer et al., 2010) had the goal of annotating all sentences of cabinet protocols with its speaker. They used a rule-based approach exploiting properties of German morphology. Speaker attribution of sentences has also been studied on dialogues in literature, where again the task is to annotate sentences or utterances with speaker information, where this is not explicit. An example is He et al. (2013) who applied supervised machine learning to the task. We do the same in this study but the genre is different and our data points are usually much longer than a single sentence.

Still, the task has similarities with closed-class author attribution. A taxonomy of six feature categories has been proposed for this task by Stamatatos (2009): character, lexical, syntactic, structural, semantic, and application-specific. The first two types have the advantage that they can be computed with very little analysis of the text; they include frequency counts of function words, punctuation marks, and short ngrams. Syntactic features can refer to part-of-speech tags or ngrams of these. Structural features include word length and sentence length as well as layout features.

Features requiring detailed analysis of texts, such as full syntactic parsing and topic modelling have also been used. Zhang et al. (2014) used dependency parsing as well as morphological and syntactic features, while Savoy (2013) employed topic modelling as a basis for feature selection. Seroussi et al. (2014) showed how variants of topic modelling can be used to predict authorship and Sari (2018) used topic modelling to analyse which features are effective under different conditions, showing content-based features to be more effective when the diversity of topics in the document set is more varied.

Given a set of selected features that can be used for profiling documents as well as authors, a method is needed to decide among the authors for a given document. Well-known methods based on a selection of frequent words are Chi-Square distance (Grieve, 2007), Burrows's Delta (Burrows, 2002), and Kullback-Leibler Distance (Zhao and Zobel, 2007). All of these compute a distance metric where the author model with the smallest distance to the document model is proposed as the most likely author. Among machine learning methods k-nearest neighbours and support vector machines have been tried, often with good results.

Neural methods have also been applied, sometimes with mixed results. The best overall system at the PAN-2015 author identification task was a character-level RNN language model (Bagnall, 2015), while the neural systems at the cross-domain author identification task at PAN-2018 did not compete well (Kestemont et al., 2018). Most systems at that event used SVMs while the best system was an ensemble system, combining features of three kinds with logistic regression.

More recently, there have been a few examples of author attribution in which the Transformer architecture (Vaswani et al., 2017), which does automatic feature extraction, has been utilised. For example, Fabien et al. (2020) introduced BertAA, a fine-tuned BERT language model for authorship classification. In experiments, the pre-trained model was fine-tuned on three different datasets in the domains of emails, blogs and movie reviews, respectively. State-of-the-art performance was obtained on all three datasets either with plain BertAA or with BertAA with additional features.

3 Data and preprocessing

The data collected at The Riksbank have two main sections: minutes from monetary policy meetings and public speeches given by Executive Board members. The minutes are from two periods: One batch starting in February 2000 and ending in May 2007, and another beginning in June 2007 and ending in April 2018. Minutes from the earlier period are truly anonymous, while the minutes from the later period have been anonymised for the purposes of this study. An overview of the data can be found in Table 2.

The speeches have been collected during a somewhat longer period, from 1997 forward. The speeches mostly address the current economic situation and are addressed to a variety of audiences such as banks, regional authorities, chambers of commerce, and parliamentarians.

Both minutes and speeches were originally in

either doc- or PDF-format. Texts were extracted from the PDF-files using the Apache Tika parser[2] accessed via a Python port[3]. From the minutes we then used regular expressions to remove data that was not text such as multiple empty lines, page headers, pagination and table cell data.

The outline of the minutes has changed over the years but is typically divided into four numbered sections. Some minutes have less than four sections and a few of them have more. Each section is supplied with a heading that starts with an initial '§'-sign. The contributions are found in a separate section with a heading such as *Penningpolitisk diskussion*, 'Discussion on monetary policies' or just *Diskussion*. This section is the one from which we extract contributions for the experiments.

A contribution from a board member in the minutes is as a rule introduced with the member's title, e.g., *Förste vice riksbankchef*, 'First Deputy Governor' and full name. All text following this introductory phrase and lasting until a new introduction of the same type is encountered has been allocated to a single contribution. A member may speak at a meeting on several occasions and so we have also collected these together as aggregated contributions. The total number of individual contributions is 900, and the aggregated contributions amount to 385.

Data type	Numbers
Speeches	399
Meetings / Minutes	65
Members present at meetings	5-6
Members during 2007-2018	12
Individual contributions	900
min length (in tokens)	11
max length (in tokens)	2760
Aggregated contributions	385
min length (in tokens)	68
max length (in tokens)	5095
Individual contributions (BERT)	1738
min length (in tokens)	13
max length (in tokens)	512
Aggregated contributions (BERT)	1434
min length (in tokens)	32
max length (in tokens)	512

Table 2: Overview of the data used in the study.

In most meetings six members including the Governor are present. There are a few meetings with fewer members present. It does not happen that a member does not contribute to the discussion at all. Some members have been present at the majority of meetings, others at only a few e.g., because their period as director ended. The minutes and the speeches have all been parsed by the Sparv parser (Borin et al., 2016). The information obtained from Sparv includes lemmas, part-of-speech tags and word senses, which we have used in subsequent processing.

The speeches, all in edited written form, are known to be given by certain members. All text of a speech, with the exception of some metadata information supplied in the header, has been kept. The main processing of the speeches is word based (frequency counts, topic modelling) and for this reason, we did not clean them to the same extent as the minutes.

For fine-tuning the pre-trained BERT model, we used the raw texts from the minutes as data (the speeches were not used in this setting), masking titles, names and gendered pronouns. The masking was done assuming such information could steer the model towards certain predictions, trivialising the task and hampering generalisation to the truly anonymous setting where this information is absent.

For both the individual and the aggregated data, the length of the contributions varies significantly. As seen in Table 2, the aggregated contributions range from 68 to 5095 tokens. Due to the limitations of BERT handling long text sequences, this posed a problem. Other architectures, such as the Longformer (Beltagy et al., 2020), have been proposed to mitigate this problem. However, in Swedish, BERT is currently the best option. What we did in our experiments, was to chunk the long texts into several smaller texts. This was done by adding up sentences of a text until the addition of one more sentence would yield a text with more than 512 tokens.

4 Features used in the experiments

We have framed our problem as a closed set classification task and applied a number of different methods. Burrows' Delta uses lexical features, which are detailed below, in Section 5.1, and the BERT model uses its own feature selection. However, for the SVM and MLP models, we have investigated various properties with the potential to differentiate between members. For each of the properties, one or more features were defined. The focus is on properties and features that relate to content and application.

[2] https://tika.apache.org/
[3] https://github.com/chrismattmann/tika-python

In the rest of this section we motivate the choice of features.

4.1 Topic modelling

We assume that the topics members address in their speeches are more or less the same as those they address in meetings as they have different backgrounds, affiliations and areas of expertise. We used lemmatized content words for the topic modelling, where we defined a content word as a word with one of the part-of-speech tags adjective, adverb, foreign word, noun, proper noun, and verb as decided by the Sparv parser. Further filtering was made by applying a frequency threshold and a threshold for spread. We trained multiple topic models with different hyperparameters, we used the NPMI coherence measure (Röder et al., 2015) that estimates coherence among word pairs in a topic based on their pairwise associations, as guide to the final topic model.

After a number of trials we found that the full data set of speeches could best be captured by eleven topics. Each topic constitutes a feature of its own. As a form of evaluation, we asked two researchers at the Riksbank to suggest short descriptions of the topics, based on the ten most probable terms for each topic. Although a few of the topics were more difficult than the others to describe convincingly, they ended up with reasonable descriptions for all of them, shown in Table 3. The fact that the topics are varied and interpretable suggests to us that the model has merits.

4.2 Sentiment analysis

Some members may have an overall negative outlook on the economy and/or the proposals discussed in board meetings, while others have a more positive one. We capture this aspect via sentiment analysis, where sentiments from the speeches are compared to sentiments expressed at meetings.

For sentiment analysis we have used a Swedish version of Vader[4] (Hutto and Gilbert, 2014) that also considers a word's sense. Vader is a lexicon and rule-based sentiment analyser. The lexicon in English Vader comprises 5500 lexical entries with sentiment scores between +5 and -5. We used the Swedish SenSALDO 0.2 sentiment lexicon (Rouces et al., 2019) with sentiment scores -1, 0 and +1, that comprises 12287 lexical entries of which 8893 are unique words. It has an accuracy of 0.89 (Rouces et al., 2019). Word sense disambiguation with the SenSALDO 0.2 lexicon is achieved using the Sparv parsed texts.

Vader also uses booster words, such as *amazingly*, to further refine the sentiment analysis. The booster dictionary used in our analyses is a slightly enhanced version of the Swedish dictionary used for sentiment analysis of consumer support e-mail conversations and comprises 89 items (Borg and Boldt, 2020). That version of Vader uses a smaller lexicon, the Swedish sentiment lexicon (Nusko et al., 2016). It was evaluated showing an 88% correspondence with human annotators.

Vader produces a compound score for each sentence, by summing the valence scores of the words according to their identified sense and normalise this sum to be between -1 (most negative) and +1 (most positive). This gives one feature. We also calculated the amount of positive, negative or neutral sentences yielding another three features. For this, we use the recommendations that a sentence has positive sentiment if the compound score is ≥ 0.05, neutral if the compound score is between -0.05 and 0.05 and negative if it is ≤ -0.05[4].

4.3 Application-specific features

Some members use more words than others. We capture this aspect by counting the number of words that each member uses, and by computing a member's share of words at a meeting. The relative share of a member's contribution gives a single feature. We also assume that the speaking order that is reported in the minutes reflects the actual speaking order at the meeting. If this order is dependent on the board member's status, or role, it could be fairly stable over time, or only change gradually. This aspect gives rise to six features corresponding to being the first speaker, the second speaker, and so on.

It is known for each member whether they have entered a reservation against the majority decision. We assume that members may differ in their incidence of entering reservations. The probability of entering a reservation is used as a feature.

4.4 Feature selection

Table 4 shows the properties we have investigated. Topic distribution and Sentiments cover the contents of contributions while the rest are application-specific capturing aspects of members' meeting behaviour. For each property, we first determined whether it could have some predictive value on its

[4] https://github.com/cjhutto/vaderSentiment

Topic	Description	Most probable terms
0	Monetary policies general	*styrränta, inflationsförväntning, inflationspolitik, mena, nominell* policy rate, expectation on inflation, inflation targeting, mean, nominal
1	Housing and private debt	*skuldsättning, skuld, bostadspris, bostad, bostadsmarknad* indebtness, debt, price of housing, housing, housing market
2	Financial stability and macro prudential	*myndighet, verktyg, institut, makrotillsyn, regelverk* public authority, tools, institute, macro supervision, regulations
3	Public debt and quantitative easing	*balansräkning, obligation, statsobligation, avkastning, miljard* balance sheet, bond, gobernment bond, returns, billion
4	Transparency and communication	*direktion, möte, öppenhet, prisnivå, kommunikation* Executive board, meeting, transparency, price level, communication
5	Labour market	*arbetsmarknad, produktivitet, vänta, inflationsförväntning, inflationsrapport* labour market, productivity, wait, expectation on inflation, inflation report
6	Monetary policy general II	*tillgångspris, resursutnyttjande, inflationsmålspolitik, mena, nominell* asset price, resource utilization, inflation targeting, mean, nominal
7	International trade euro area	*euro, eu, emu, konkurrens, handel* Euro, EU, EMU, competition, trade
8	International trade general	*offentlig, sparande, diagram, bytesbalans, export* public, savings, diagram, balance of payments, export
9	Payment system	*betalning, pengar, kontanter, betalningssystem, infrastruktur* payment, money, cash, payment system, infrastructure
10	Inflation targeting and the policy rate path	*resursutnyttjande, diagram, räntebana, stabilisera, hållbar* resource utilization, diagram, policy rate path, stabilize, sustainable

Table 3: Descriptions of the produced topics with the five most probable terms.

own using both MLP- and SVM-systems[5]. It can be seen from Table 4 that all selected properties give performance above a random baseline which, for six participants present in each meeting, would give a theoretical accuracy of 16.7%. Topic distribution is by far the property that has the best results.

In total, our feature set consists of 37 features. Since we are interested in how these features impact member classification, we employed two different feature selection methods. The first approach is a Recursive Feature Elimination (RFE) which is able to find a set of features that carry the most predictive power. The second is based on a Python implementation[6] of the Boruta algorithm (Kursa and Rudnicki, 2010). The rationale behind using Boruta is the algorithm's ability to provide a set of relevant features, contrary to the minimal optimal feature sets provided by for example RFE. This means that we with Boruta are able to get a set of all features that have some impact on the prediction, while with RFE we can choose to extract the N most important features. By using a combination of these algorithms, we can therefore gain knowledge about which features carry the most predictive power if we wanted to slim down the classification model, but also a picture of which of the features provide at least some information for the classification task.

5 Experiments

This section elaborates on the details of the different systems and their performance. All results are shown in Table 5.

5.1 A traditional system: Burrows' Delta

For comparison, we tested an implementation of Burrows' Delta under different conditions. Three different feature sets were used, one relying solely on the most frequent words in the corpus of speeches, another where proper nouns were removed, as these include references to the speaker we wish to identify, and a third relying on the most frequent content words, where a content word was defined as a noun, verb or adjective. Following Evert et al. (2015) we also looked at the effect of normalising the feature vectors and compared two different measures: Manhattan distance and Cosine similarity.

Initial tests were made on a corpus where all contributions from one member had been collected into one text yielding a total of 12 texts. These suggested that the frequency-based features gave slightly better results than the other two, with 7 out of 12 members being predicted correctly, and 9 out of 12 being included in the two first predictions. This selection of features was then used for predicting the speaker of contributions at meetings. The number of features was also varied showing clear

[5] See section 5.2 for a description of the experimental setup

[6] https://github.com/scikit-learn-contrib/boruta_py

Property	Features	Accuracy (SVM)	Accuracy (MLP)
Length (absolute)	1	30.57%	26.67%
Length (relative)	1	23.19%	28.33%
Order (only position)	1	25.13%	21.25%
Order (probabilities)	6	42.31%	40.97%
Reservation	1	23.10%	21.25%
Sentiments (compound)	1	18.77%	16.45%
Sentiments (ratios)	3	23.11%	16.06%
Topic distribution	11	63.70%	62.84%
Burrows Delta	12	24.86%	29.34%

Table 4: Properties used and their performance as single predictors for the SVM and MLP models.

improvements from 300 features upwards with a peak around 500. Using normalised feature vectors and cosine distance consistently gave better performance by two or more points. The best results are reported in Table 5.

We observe that the best result for the aggregated contributions is close to that for the topic models. We also see that results drop when predicting speakers of individual contributions but are still far above chance. Adding more features does not generally improve predictions.

We can also note that the performance of using the Burrows' Delta models for different speakers to generate features to be included in an SVM-classifier differs greatly from using the standalone system for classifying members with Burrows' Delta.

5.2 The SVM and MLP systems

Each type of feature was first tested individually to see whether it could beat a random baseline. The results are reported in Table 4.

The systems, written in Python, use the scikit-learn library (Pedregosa et al., 2011), with the implementations of support vector machines (SVC) and multilayer perceptrons (MLPClassifier) as the algorithms for the classification task.

We used a 5-fold cross-validation procedure to randomly split the data into training and test data. Since we wanted to do the prediction of the contributing members on a meeting level, we let the individual meetings be the unit assigned to either the train or test portion of each fold, with all member contributions extracted from the particular meeting. In the cross-validation procedure, it is however customary to balance the classes (in this case, the members) evenly across all folds, but as a consequence of the importance of keeping the integrity of each meeting, it was not possible to achieve a perfect balance of classes in all folds.

In each training fold, we performed a second 5-fold cross-validation procedure to optimise the hyperparameters of the selected classification model. For the SVM, we optimised the C and gamma values with a radial basis function (RBF) kernel. The MLP was optimised with its hidden layer sizes and the L2-regularisation term (named alpha in scikit-learn) for the *Limited Memory Broyden–Fletcher–Goldfarb–Shanno* (lbfgs) solver.

We implemented a custom prediction step with two restrictions for the classification task, namely, for each meeting;

- Only members present in the meeting can be predicted.
- A member can only be predicted once per meeting.

5.3 Ensemble systems

Using the same features and the two restrictions just described, two ensemble systems of SVMs and MLPs were implemented[7]. The first is a soft voting system, where both an SVM and an MLP are trained as described in section 5.2. At the prediction step the classification probabilities, for each possible board member, of both the SVM and MLP are added together and averaged between the two classifiers. The board member with the highest average probability is then subsequently selected as the classifier output for the given meeting. Since we noticed subtle differences in how the SVM and MLP predicted certain meetings, the rationale behind this approach was to try to make a more robust prediction, leveraging the strengths of both classifiers.

The second ensemble system is a hybrid of an MLP and an SVM, following the method used in Garg et al. (2021). The system consists of an MLP that is trained on the training splits in a regular fashion, but whose weight matrix from the final

[7]The cross-validation and hyperparameter optimisation were performed in the same fashion as described in section 5.2

hidden layer is used as features by an additional SVM classifier.

5.4 BERT-based system

As with the SVM and MLP systems, we used a 5-fold cross-validation procedure to randomly split the data into training and test data. The fine-tuning procedure was implemented using Transformers (Wolf et al., 2020) and PyTorch (Paszke et al., 2019). To make this method comparable with the other methods described, we combined the predictions for smaller chunks of a given contribution into one single prediction. Thus, we had to keep track of the contribution ID when splitting into training and test data and make sure that all smaller chunks for a given contribution were in the same partition. The combining was then done by summing the raw output scores from the model for all chunks of a given contribution before picking the class with the highest score as the prediction. This way, we got a single prediction for each contribution.

For both the aggregated and individual data, we did experiments of two kinds, one where only the members present at a particular meeting were considered when aggregating predictions and one that disregarded the notion of meetings. The former setting is similar to the setting used for the SVM and MLP systems, with the only difference being that each member in a meeting could now be predicted multiple times. In the latter case, no information about what members participated at a particular meeting was given to the model. Thus, the model had to predict the member from the pool of all 12 members. Surprisingly, at the end of training for each fold, the results were exactly the same in all cases but one where the first approach had an increase in accuracy of approximately 1% compared to the second approach. This effect was seen in both the aggregated and individual data.

In each fold, the data was prepared as input to the BERT model by retrieving input ids, and the attentions mask for each batch of sequences. A batch size of 8 was used, and the model was fine-tuned for 10 epochs on a Tesla P100-PCIE-16GB GPU with a learning rate of 10^{-5}. 10 epochs seemed to be suitable for this problem and dataset, as loss converged without causing overfitting.

6 Results

The results for classification accuracy of all tested systems are presented in Table 5. For all systems, as expected, the aggregated contributions score higher than the individual contributions. Classifying aggregated (and longer) contributions are naturally a less complex problem compared to individual contributions due to the reduced number of classifications to be performed per meeting. It should however be noted that the tested systems, especially the system based on BERT, are able to handle this change of scope in an acceptable manner considering the increased task complexity.

System	**Contributions**	
	Aggregated	**Individual**
Delta, standard, 500feats	55.54%	33.89%
Delta, normalised, 500feats	60.33%	42.41%
SVM (*RFE*)	78.20%	54.18%
SVM (*Boruta*)	79.70%	57.55%
SVM (*All*)	78.56%	56.98%
MLP (*RFE*)	76.22%	52.45%
MLP (*Boruta*)	77.15%	54.15%
MLP (*All*)	74.66%	54.55%
Soft voting (*RFE*)	77.25%	54.50%
Soft voting (*Boruta*)	77.95%	55.83%
Soft voting (*All*)	76.48%	57.52%
Hybrid (*RFE*)	78.71%	54.78%
Hybrid (*Boruta*)	78.50%	54.66%
Hybrid (*All*)	79.31%	55.70%
BERT	94.81%	83.78%

Table 5: Results of the classification accuracy for different systems, feature sets and types of member contributions.

Furthermore, the results indicate that the SVM and MLP classification methods performed significantly better than the random baseline and that the differences between these methods were relatively small. When including all the features listed in Table 4, we saw a lower classification performance for all feature-based systems, compared to when we included only a subset of the features (see Table 5). The best results were generally found with the subset of features selected by the Boruta-algorithm, referred to as *Boruta* in Table 5. The best performance of any feature based system can be seen in the standalone SVM system with an accuracy of 79.70% on the aggregated data and boruta feature selection. The best performance of the Ensemble systems were found in the hybrid system with an accuracy of 79.31%, followed by the soft voting system with an accuracy of 77.95%. The standalone MLP system performed the generally lowest scores, with the highest being 77.15%,

An even smaller subset of features (the *RFE* feature subset), including the 10 features with the most predictive power according to the Recursive Feature Elimination, were able to perform almost on

par with the other feature sets. This also aligns with what was seen when each feature was tested individually (see Table 4), where some of the features scored close to the random baseline. The features present in the subset created with *RFE*, **topics**, **length (absolute)**, and **speaking order**, were also some of the highest performing individual features. It should however be noted that not all of the topics are included among the top 10. Three of the topics are not (topic 6, topic 7, and topic 9). These topics were also some of the few features that most often were omitted as features by the Boruta algorithm. All this taken into consideration, we can conclude that a small number of features carry great predictive power for the classification task.

The fine-tuned BERT model obtained 94.81% accuracy for all folds combined on the aggregated data, and an accuracy of 83.78% on the individual data. Since the data used for the BERT model had to be split into smaller chunks to fit the input limit of BERT, we tried the same chunking approach for all the non-BERT systems. Since some of the aggregated contributions were fairly long (see Table 2) the total number of contributions increased significantly, while also rendering the restriction of only being able to predict a member once per meeting less effective. The SVM, MLP, and ensemble systems did therefore perform worse with this data chunking approach, resulting in accuracies between 55-58% on the aggregated data.

7 Conclusions and discussion

In this work, the main purpose has been to investigate a set of interpretable features for identifying speakers from minutes. With the aid of feature selection algorithms, we are able to pin down the most important features, while also excluding some of the less relevant, and simultaneously improve the classification performance. Topic models generated from speeches given by directors of the board of The Riksbank turned out to be a good predictor of what they say in board meetings. Combined with other features such as wordiness, speaking order and sentiment analysis we could reach an accuracy close to 80% in predicting which director said what. Not surprisingly the fine-tuned BERT model has the best performance in predicting which board member made a certain contribution. This is in line with the performance of similar models in other attribution tasks (cf. Fabien et al. (2020)) and points to Transformer-based models being good feature extractors. While we have only investigated one corpus of minutes, the methods we've tried have a wider application; similar types of meetings and minutes are common in financial and other public institutions where transparency and accountability is an issue.

The success of the BERT-model suggests that members are consistent in their argumentation across meetings. An interesting aspect is the fact that the minutes of the meetings are not written by the members, which should make this task harder than standard author attribution. Given this, we find that the BERT model provides a strong benchmark for de-anonymisation of minutes.

The BERT-model, unlike the features used for the other models, is not easily interpretable. Yet, as new techniques for interpreting models such as BERT are emerging, we would like to investigate what the BERT-model actually considers when making predictions. For example, whether it looks at stylistic features in how the minute taker writes about a particular member or features more related to the content and topics of the contributions.

The properties, coupled with analysis of the overall differences of the minutes under the two conditions, are likely to be helpful in future research on de-anonymising the minutes from the earlier period. Although members are not referred to by name there is a similar structure to the minutes and the discussions so that contributions can be identified. While performance may be lower for all models when applied to minutes for the earlier period, the data obtained from the non-anonymous minutes could then be used for training. For example, we know from confusion matrices which speaker models are often confused.

There are features we have not yet investigated such as members' style of argumentation, or rhetorical structure, which potentially could be helpful. As we now also have identified the topics discussed during the meetings, we can analyse members' attitudes, i.e. sentiments, towards each topic. This can also be included in our model. An analysis of the parse trees of contributions could also yield features at a more fine-grained level than topics, such as individual members' hobby-horses.

References

Samanvaya Agarwala, Saipriya Kamathb, Krishnamurthy Subramanianc, and Prasanna Tantrid. 2022.

Board conduct in banks. *Journal of Banking and Finance*, 138.

Douglas Bagnall. 2015. Author identification using multi-headed recurrent neural networks. *CoRR*, abs/1506.04891.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Anton Borg and Martin Boldt. 2020. Using VADER sentiment and SVM for predicting customer response sentiment. *Expert Systems with Applications*, 162:113746.

Lars Borin, Markus Forsberg, Martin Hammarstedt, Dan Rosén, Roland Schäfer, and Anne Schumacher. 2016. Sparv: Språkbanken's corpus annotation pipeline infrastructure. In *SLTC 2016. The Sixth Swedish Language Technology Conference, Umeå University, 17-18 November, 2016*.

John F. Burrows. 2002. Delta: A measure of stylistic difference and a guide to likely authorship. *Literary and Linguistic Computing*, 17:267–287.

Stefan Evert, Thomas Proisi, Fotis Jannidis, Steffen Pielström, Christof Schöch, and Thorsten Vitt. 2015. Towards a better understanding of burrows's delta in literary authorship attribution. In *Proceedings of NAACL-HLT Fourth Workshop on Computational Linguistics for Literature*, pages 79–88, Denver, Colorado.

Maël Fabien, Esau Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. 2020. BertAA: BERT fine-tuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 127–137, Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLPAI).

Armin Falk and Florian Zimmermann. 2018. Information processing and commitment. *Economic Journal*, 128:1983–2002.

Ginni Garg, Dheeraj Kumar, Yash Sonker, Ritu Garg, et al. 2021. A hybrid MLP-SVM model for classification using spatial-spectral features on hyper-spectral images. *arXiv preprint arXiv:2101.00214*.

Jack Grieve. 2007. Quantitative author attribution. *Literary and Linguistic Computing*, 22:251–270.

Stephen Hansen, Michael McMahon, and Andrea Prat. 2018. Transparency and deliberation within the FOMC: A computational linguistics approach. *The Quarterly Journal of Economics*, pages 801–870.

Hua He, Denilson Barbosa, and Grzegorz Kondrak. 2013. Identification of speakers in novels. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1312–1320, Sofia, Bulgaria. Association for Computational Linguistics.

Bengt Holmström. 1999. Managerial incentive problems: A dynamic perspective. *Review of Economic Studies: Special Issue: Contracts*, 66(1):169–182.

Clayton J. Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI*.

Mike Kestemont, Michael Tschuggnall, Efstathios Stamatatos, Walter Daelemans, Günther Specht, Benno Stein, and Martin Potthast. 2018. Overview of the author identification task at pan-2018: Cross-domain authorship attribution and style change detection. In *CEUR Workshop Proceedings*, volume 2125.

Miron B Kursa and Witold R Rudnicki. 2010. Feature selection with the boruta package. *Journal of statistical software*, 36:1–13.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the national library of Sweden – making a Swedish BERT. *arXiv preprint arXiv:2007.01658*.

Bianka Nusko, Nina Tahmasebi, and Olof Mogren. 2016. Building a sentiment lexicon for swedish. *Linköping Electronic Conference Proceedings*, 126(006):32—-37.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct):2825–2830.

Jacobo Rouces, Nina Tahmasebi, Lars Borin, and Stian Rødven Eide. 2019. Sensaldo: Creating a sentiment lexicon for Swedish. *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pages 4192–4198.

Josef Ruppenhofer, Caroline Sporleder, and Fabian Shirokov. 2010. Speaker attribution in cabinet protocols. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence

measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM '15)*, WSDM '15, page 399–408, New York, NY, USA. Association for Computing Machinery.

Yunita Sari. 2018. *Neural and non-neural approaches to authorship attribution*. Ph.D. thesis, University of Sheffield.

Jacques Savoy. 2013. Authorship attribution based on a probabilistic topic model. *Information Processing and Management*, 49(1):341–354.

Miriam Schwartz-Ziv and Michael S. Weisbach. 2013. What do boards really do? evidence from minutes of board meetings. *Journal of Financial Economics*, 108(2):349—366.

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2014. Authorship attribution with topic models. *Computational Linguistics*, 40(2):269–310.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chunxia Zhang, Xindong Wu, Zhendong Niu, and Wei Ding. 2014. Authorship identification from unstructured text. *Knowledge-Based Systems*, 66:99–111.

Ying Zhao and Justin Zobel. 2007. Entropy-based authorship search in large document collection. In *Proceedings of the 29th European Conference on IR Research, ECIR*, pages 381–392.

On the Concept of Resource-Efficiency in NLP

Luise Dürlich[*1,2] **Evangelia Gogoulou**[*1,3] **Joakim Nivre**[1,2]

[1]RISE Research Institutes of Sweden, Department of Computer Science
[2]Uppsala University, Department of Linguistics and Philology
[3]KTH Royal Institute of Technology, Division of Software and Computer Systems

{luise.durlich, evangelia.gogoulou, joakim.nivre}@ri.se

Abstract

Resource-efficiency is a growing concern in the NLP community. But what are the resources we care about and why? How do we measure efficiency in a way that is reliable and relevant? And how do we balance efficiency and other important concerns? Based on a review of the emerging literature on the subject, we discuss different ways of conceptualizing efficiency in terms of product and cost, using a simple case study on fine-tuning and knowledge distillation for illustration. We propose a novel metric of amortized efficiency that is better suited for life-cycle analysis than existing metrics.

1 Introduction

Resource-efficiency has recently become a more prominent concern in the NLP community. The Association for Computational Linguistics (ACL) has issued an Efficient NLP Policy Document[1] and most conferences now have a special track devoted to efficient methods in NLP. The major reason for this increased attention to efficiency can be found in the perceived negative effects of scaling NLP models (and AI models more generally) to unprecedented sizes, which increases energy consumption and carbon footprint as well as raises barriers to participation in NLP research for economic reasons (Strubell et al., 2019; Schwartz et al., 2020). These considerations are important and deserve serious attention, but they are not the only reasons to care about resource-efficiency. Traditional concerns like guaranteeing that models can be executed with sufficient speed to enable real-time processing, or with sufficiently low memory footprint to fit on small devices, will continue to be important as well.

Resource-efficiency is however a complex and multifaceted problem. First, there are many relevant types of resources, which interact in complex (and sometimes antagonistic) ways. For example, adding more computational resources may improve time efficiency but increase energy consumption. For some of these resources, obtaining relevant and reliable measurements can also be a challenge, especially if the consumption depends on both software and hardware properties. Furthermore, the life-cycle of a typical NLP model can be divided into different phases, like pre-training, fine-tuning and (long-term) inference, which often have very different resource requirements but nevertheless need to be related to each other in order to obtain a holistic view of total resource consumption. Since one and the same (pre-trained) model can be fine-tuned and deployed in multiple instances, it may also be necessary to amortize the training cost in order to arrive at a fair overall assessment.

To do justice to this complexity, we must resist the temptation to reduce the notion of resource-efficiency to a single metric or equation. Instead, we need to develop a conceptual framework that supports reasoning about the interaction of different resources while taking the different phases of the life-cycle into account. The emerging literature on the subject shows a growing awareness of this need, and there are a number of promising proposals that address parts of the problem. In this paper, we review some of these proposals and discuss issues that arise when trying to define and measure efficiency in relation to NLP models.[2] We specifically address the need for a holistic assessment of efficiency over the entire life-cycle of a model and propose a novel notion of amortized efficiency. All notions and metrics are illustrated in a small case study on fine-tuning and knowledge distillation.

[*]Equal contribution to this work.

[1]https://www.aclweb.org/portal/content/efficient-nlp-policy-document

[2]Most of the discussion is relevant also to other branches of AI, although some of the examples and metrics discussed are specific to NLP.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 135–145

2 Related Work

Strubell et al. (2019) were among the first to discuss the increasing resource requirements in NLP. They provide estimates of the energy needed to train a number of popular NLP models, including T2T (Vaswani et al., 2017), ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and GPT2 (Radford et al., 2019). Based on those estimates, they also estimate the cost in dollars and the CO_2 emission associated with model training. In addition to the cost of training a single model, they provide a case study of the additional (much larger) costs involved in hyperparameter tuning and model fine-tuning. Their final recommendations include: (a) Authors should report training time and sensitivity to hyperparameters. (b) Academic researchers need equitable access to computation resources. (c) Researchers should prioritize computationally efficient hardware and algorithms.

Schwartz et al. (2020) note that training costs in AI increased 300,000 times from 2012 to 2017, with costs doubling every few months, and argue that focusing only on the attainment of state-of-the-art accuracy ignores the economic, environmental, or social cost of reaching the reported accuracy. They advocate research on *Green AI* – AI research that is more environmentally friendly and inclusive than traditional research, which they call *Red AI*. Specifically, they propose making *efficiency* a more common evaluation criterion for AI papers alongside accuracy and related measures.

Hershcovich et al. (2022) focus specifically on environmental impact and propose a climate performance model card that can be used with only limited information about experiments and underlying computer hardware. At a minimum authors are asked to report (a) whether the model is publicly available, (b) how much time it takes to train the final model, (c) how much time was spent on all experiments (including hyperparameter search), (d) what the total energy consumption was, and (e) at which location the computations were performed. In addition, authors are encouraged to report on the energy mix at the location and the CO_2 emission associated with different phases of model development and use.

Liu et al. (2022) propose a new benchmark for efficient NLP models called ELUE (Efficient Language Understanding Evaluation) based on the concept of Pareto state of the art, which a model is said to achieve if it achieves the best performance at a given cost level. The cost measures used in ELUE are number of model parameters and number of floating point operations (FLOPs), while performance measures vary depending on the task (sentiment analysis, natural language inference, paraphrase and textual similarity).

Treviso et al. (2022) provide a survey of current research on efficient methods for NLP, using a taxonomy based on different aspects or phases of the model life-cycle: data collection and preprocessing, model design, training (including pre-training and fine-tuning), inference, and model selection. Following Schwartz et al. (2020), they define efficiency as the cost of a model in relation to the results it produces. They observe that cost can be measured along multiple dimensions, such as computational, time-wise or environmental cost, and that using a single cost indicator can be misleading. They also emphasize the importance of separately characterizing different stages of the model life-cycle and acknowledge that properly measuring efficiency remains a challenge.

Dehghani et al. (2022) elaborate on the theme of potentially misleading efficiency characterizations by showing that some of the most commonly used cost indicators – number of model parameters, FLOPs, and throughput (msec/example) – can easily contradict each other when used to compare models and are therefore insufficient as standalone metrics. They again stress the importance of distinguishing training cost from inference cost, and point out that their relative importance may vary depending on context and use case. For example, training efficiency is crucial if a model needs to be retrained often, while inference efficiency may be critical in embedded applications.

3 The Concept of Efficiency in NLP

Efficiency is commonly defined as the ratio of useful output to total input:[3]

$$r = \frac{P}{C} \tag{1}$$

where P is the amount of useful output or results, the *product*, and C is the total *cost* of producing the results, often defined as the amount of resources consumed. A process or system can then be said

[3]Historically, the technical concept of efficiency arose in engineering in the nineteenth century, in the analysis of engine performance (thermodynamic efficiency); it was subsequently adopted in economy and social science by Vilfredo Pareto and others (Mitcham, 1994).

to reach maximum efficiency if a specific desired result is obtained with the minimal possible amount of resources, or if the maximum amount of results is obtained from a given resource. More generally, maximum efficiency holds when it is not possible to increase the product without increasing the cost, nor reduce the cost without reducing the product.

In order to apply this concept of efficiency to NLP, we first have to decide what counts as useful output or results – the product P in Equation 1. We then need to figure out how to measure the cost C in terms of resources consumed. Finally, we need to come up with relevant ways of relating P to C in different contexts of research, development and deployment, as well as aggregating the results into a life-cycle analysis. We will begin by discussing the last question, because it has a bearing on how we approach the other two.

3.1 The Life-Cycle of an NLP Model

It is natural to divide the life-span of an NLP model into two phases: *development* and *deployment*. In the development phase, the model is created, optimized and validated for use. In the deployment phase, it is being used to process new text data in one or more applications. The development phase of an NLP model today typically includes several stages of training, some or all of which may be repeated multiple times in order to optimize various hyperparameters, as well as validation on held-out data to estimate model performance. The deployment phase is more homogeneous in that it mainly consists in using the model for inference on new data, although this may be interrupted by brief development phases to keep the model up to date.

As researchers, we naturally tend to focus on the development of new models and many models developed in a research context may never enter the deployment phase at all. Since the development phase is typically also more computationally intensive than the deployment phase, it is therefore not surprising that early papers concerned with the increasing energy consumption of NLP research, such as Strubell et al. (2019) and Schwartz et al. (2020), mainly focused on the development phase. Nevertheless, for models that are actually put to use in large-scale applications, resources consumed during the deployment phase may in the long run be much more important, and efficiency in the deployment phase is therefore an equally valid concern. This is also the focus of the recently proposed evaluation framework ELUE (Liu et al., 2022).

As will be discussed in the following sections, some proposed efficiency metrics are better suited for one of the two phases, although they can often be adapted to the other phase as well. However, the question is whether there is also a need for metrics that capture the combined resource usage at development and deployment, and how such metrics can be constructed. One reason for being interested in combined metrics is that there may be trade-offs between resources spent during development and deployment, respectively, so that spending more resources in development may lead to more efficient deployment (or vice versa). To arrive at a more holistic assessment of efficiency, we need to define efficiency metrics for deployment that also incorporate development costs. Before we propose such a metric, we need to discuss how to conceptualize products and costs of NLP models.

3.2 The Products of an NLP Model

What is the output that we want to produce at the lowest possible cost in NLP? Is it simply a model capable of processing natural language (as input or output or both)? Is it the performance of such a model on one or more NLP tasks? Or is it the actual output of such a model when processing natural language at a certain performance level? All of these answers are potentially relevant, and have been considered in the literature, but they give rise to different notions of efficiency and require different metrics and measurement procedures.

Regarding the model itself as the product is of limited interest in most circumstances, since it does not take performance into account and only makes sense for the development phase. It is therefore more common to take model performance, as measured on some standard benchmark, as a relevant product quantity, which can be plotted as a function of some relevant cost to obtain a so-called Pareto front (with corresponding concepts of Pareto improvement and Pareto state of the art), as illustrated in Figure 1, reproduced from Liu et al. (2022).

One advantage of the product-as-performance model is that it can be applied to the deployment phase as well as the development phase, although the cost measurements are different in the two cases. For the development phase, we want to measure the *total* cost incurred to produce a model with a given performance, which depends on a multitude of factors, such as the size of the model, the num-

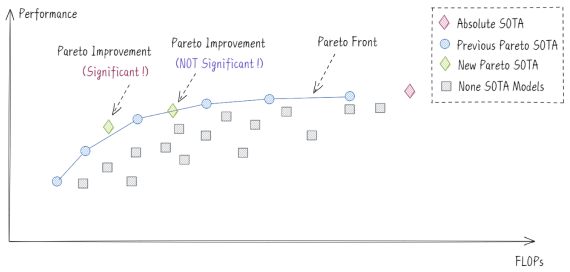


Figure 1: Pareto front with model performance as the product and cost measured in FLOPs (Liu et al., 2022).

ber of hyperparameters that need to be tuned, and the data efficiency of the learning algorithm. For the deployment phase, we instead focus on the *average* cost of processing a typical input instance, such as a natural language sentence or a text document, independently of the development cost of the model. Separating the two phases in this way is perfectly adequate in many circumstances, but the fact that we measure total cost in one case and average cost in the other makes it impossible to combine the measurements into a global life-cycle analysis. To overcome this limitation, we need a notion of product that is not defined (only) in terms of model performance but also considers the actual output produced by a model.

If we take the product to be the amount of data processed by a model in the deployment phase, then we can integrate the development cost in the efficiency metric as a debt that is amortized during deployment. Under this model, the average cost of processing an input instance is not constant but decreases over the life-time of a model, which allows us to capture possible trade-offs between development and deployment costs. For example, it may sometimes be worth investing more resources into the development phase if this leads to a lower development cost in the long run. Moreover, this model allows us to reason about how long a model needs to be in use to "break even" in this respect.

An important argument against the product-as-output model is that it is trivial (but uninteresting) to produce a maximally efficient model that produces random output. It thus seems that a relevant life-cycle analysis requires us to incorporate both model performance and model output into the notion of product. There are two obvious ways to do this, each with its own advantages and drawbacks. The first is to stipulate a minimum performance level that a model must reach to be considered valid and to treat all models reaching this threshold as ceteris paribus equivalent. The second way is to use the performance level as a weighting function when calculating the product of a model. We will stick to the first and simpler approach in our case study later, but first we need to discuss the other quantity in the efficiency equation – the cost.

3.3 The Costs of an NLP Model

Schwartz et al. (2020) propose the following formula for estimating the computational cost of producing a result R:

$$Cost(R) \propto E \cdot D \cdot H \tag{2}$$

where E is the cost of executing the model on a single example, D is the size of the training set (which controls how many times the model is executed during a training run), and H is the number of hyperparameter experiments (which controls how many times the model is trained during model development). How can we understand this in the light of the previous discussion?

First, it should be noted that this is not an exact equality. The claim is only that the cost is proportional to the product of factors on the right hand side, but the exact cost may depend on other factors that may be hard to control. Depending on what type of cost is considered – a question that we will return to below – the estimate may be more or less exact. Second, the notion of a *result* is not really specified, but seems to correspond to our notion of *product* and is therefore open to the same variable interpretations as discussed in the previous section. Third, as stated above, the formula applies only to the development phase, where the result/product is naturally understood as the performance of the final model. To clarify this, we replace R (for result) with P_P (for product-as-performance) and add the subscript T (for training) to the factors E and D:

$$DevCost(P_P) \propto E_T \cdot D_T \cdot H \tag{3}$$

Schwartz et al. (2020) go on to observe that a formula appropriate for inference during the deployment phase can be obtained by simply removing the factors D and H (and, in our new notation, changing E_T to E_I since the cost of processing a single input instance is typically not the same at training and inference time):

$$DepCost(P_P) \propto E_I \tag{4}$$

This corresponds to the product-as-performance model for the deployment phase discussed in the previous section, based on the average cost of processing a typical input instance, and has the same limitations. It ignores the quantity of data processed by a model, and it is insensitive to the initial investment in terms of development cost. To overcome the first limitation, we can add back the factor D, now representing the amount of data processed during deployment (instead of the amount of training data), and replace product-as-performance (P_P) by product-as-output (P_O):

$$DepCost(P_O) \propto E_I \cdot D_I \quad (5)$$

To overcome the second limitation, we have to add the development cost to the equation:

$$DepCost(P_O) \propto E_T \cdot D_T \cdot H + E_I \cdot D_I \quad (6)$$

This allows us to quantify the product and cost as they develop over the lifetime of a model, and this is what we propose to call *amortized* efficiency based on total deployment cost, treating development cost as a debt that is amortized during the deployment phase. Our notion of amortized efficiency is inspired by the notion of amortized analysis from complexity theory (Tarjan, 1985), which averages costs over a sequence of operations. Here we instead average costs over different life-cycle phases.

As already noted, the product-as-output view is only meaningful if we also take model performance into account, either by stipulating a threshold of minimal acceptable performance or by using performance as a weight function when calculating the output produced by the model. Note, however, that we can also use the notion of total deployment cost to compare the Pareto efficiency of different models at different points of time (under a product-as-performance model) by computing average deployment cost in a way that is sensitive to development cost and lifetime usage of a model.

The discussion so far has focused on how to understand the notion of efficiency in NLP by relating different notions of *product* to an abstract notion of *cost* incurred over the different phases of lifetime of a model. However, as noted in the introduction, this abstract notion of cost can be instantiated in many different ways, often in terms of a specific resource being consumed, and it may be more or less straightforward to obtain precise measures of the resource consumption. Before illustrating the different efficiency metrics with some real data, we will therefore discuss costs and resources that have been prominent in the recent literature and motivate the selection of costs included in our case study.

Time and Space The classical notion of efficient computation from complexity theory is based on the resources of *time* and *space*. Measuring cost in terms of time and space (or memory) is important for time-critical applications and/or memory-constrained settings, but in this context we are more interested in execution time and memory consumption than in asymptotic time and space complexity. For this reason, execution time remains one of the most often reported cost measures in the literature, even though it can be hard to compare across experimental settings because it is influenced by factors such as the underlying hardware, other jobs running on the same machine, and the number of cores used (Schwartz et al., 2020). We include execution time as one of the measured costs in our case study.

Power and CO_2 Electrical power consumption and the ensuing CO_2 emission are costs that have been highlighted in the recent literature on resource-efficient NLP and AI. For example, Strubell et al. (2019) estimate the total power consumption for training NLP models based on available information about total training time, average power draw of different hardware components (GPUs, CPUs, main memory), and average power usage effectiveness (PUE) for data centers. They also discuss the corresponding CO_2 emission based on information about average CO_2 produced for power consumed in different countries and for different cloud services. Hershcovich et al. (2022) propose that climate performance model cards for NLP models should minimally include information about total energy consumption and location for the computation, ideally also information about the energy mix at the location and the CO_2 emission associated with different phases of model development and use. Against this, Schwartz et al. (2020) observe that, while both power consumption and carbon emission are highly relevant costs, they are difficult to compare across settings because they depend on hardware and local electricity infrastructure in a way that may vary over time even at the same location. In our case study, we include measurements of power consumption, but not carbon emission.

Abstract Cost Measures Given the practical difficulties to obtain exact and comparable measure-

ments of relevant costs like time, power consumption, and carbon emission, several researchers have advocated more abstract cost measures, which are easier to obtain and compare across settings while being sufficiently correlated with other costs that we care about. One such measure is model size, often expressed as number of parameters, which is independent of underlying hardware but correlates with memory consumption. However, as observed by Schwartz et al. (2020), since different models and algorithms make different use of their parameters, model size is not always strongly correlated with costs like execution time, power consumption, and carbon emission. They therefore advocate number of floating point operations (FLOPs) as the best abstract cost measure, arguing that it has the following advantages compared to other measures: (a) it directly computes the amount of work done by the running machine when executing a specific instance of a model and is thus tied to the amount of energy consumed; (b) it is agnostic to the hardware on which the model is run, which facilitates fair comparison between different approaches; (c) unlike asymptotic time complexity, it also considers the amount of work done at each time step. They acknowledge that it also has limitations, such as ignoring memory consumption and model implementation. Using FLOPs to measure computation cost has emerged as perhaps the most popular approach in the community, and it has been shown empirically to correlate well with energy consumption (Axberg, 2022); we therefore include it in our case study.

Data The amount of data (labeled or unlabeled) needed to train a given model and/or reach a certain performance is a relevant cost measure for several reasons. In AI in general, if we can make models and algorithms more data-efficient, then they will ceteris paribus be more time- and energy-efficient. In NLP specifically, it will in addition benefit low-resource languages, for which both data and computation are scarce resources.

In conclusion, no single cost metric captures all we care about, and any single metric can therefore be misleading on its own. In our case study, we show how different cost metrics can be combined with different notions of product to analyze resource-efficiency for NLP models. We include three of the most important metrics: execution time, power consumption, and FLOPs.

4 Case Study

To illustrate the different conceptualizations of resource-efficiency discussed in previous sections, we present a case study on developing and deploying a language model for a specific NLP task using different combinations of fine-tuning and knowledge distillation. The point of the study is not to advance the state of the art in resource-efficient NLP, but to show how different conceptualizations support the comparison of models of different sizes, at different performance levels, and with different development and deployment costs.

4.1 Overall Experimental Design

We apply the Swedish pre-trained language model KB-BERT (Malmsten et al., 2020) to Named Entity Recognition (NER), using data from SUCX 3.0 (Språkbanken, 2022) for fine-tuning and evaluation. We consider three scenarios:

- **Fine-tuning (FT):** The standard fine-tuning approach is followed, with a linear layer added on top of KB-BERT. The model is trained on the SUCX 3.0 training set until the validation loss no longer decreases for up to 10 epochs.
- **Task-specific distillation (TS):** We distill the fine-tuned KB-BERT model to a 6-layer BERT student model. The student model is initialized with the 6 lower layers of the teacher and then trained on the SUCX 3.0 training set using the teacher predictions on this set as ground truth.
- **Task-agnostic distillation (TA):** We distill KB-BERT to a 6-layer BERT student model using the task-agnostic distillation objective proposed by Sanh et al. (2020). Following their approach, we initialize the student with every other layer of the teacher and train on deduplicated Swedish Wikipedia data by averaging three kinds of losses for masked language modelling, knowledge distillation and cosine-distance between student and teacher hidden states. The student model is subsequently fine-tuned on the SUCX 3.0 training set with the method used in the FT experiment.

All three fine-tuned models are evaluated on the SUCX 3.0 test set. Model performance is measured using the F1 score, which is the standard evaluation metric for NER, and model output in number of

	Distillation Stage			Fine-Tuning Stage			Evaluation Stage			
	Time	**Power**	**FLOPs**	**Time**	**Power**	**FLOPs**	**Time**	**Power**	**FLOPs**	**F1**
FT	–	–	–	0:35:17	141.1	2.48×10^{16}	0:01:32	5.2	2.59×10^{15}	87.3
TS	0:18:30	77.1	1.64×10^{16}	0:35:17	141.1	2.48×10^{16}	0:01:09	3.1	1.71×10^{15}	84.9
TA	13:06:59	6848.9	3.65×10^{17}	0:18:53	74.4	1.69×10^{16}	0:01:15	3.3	1.71×10^{15}	77.6

Table 1: Performance (F1) and cost measurements (Time: hh:mm:ss, Power: Wh, FLOPs) for different stages (Distillation, Fine-tuning, Evaluation) and different development scenarios (Fine-tuning: FT, Task-specific distillation: TS, Task-agnostic distillation: TA).

tokens. We measure three different types of cost during development and deployment: execution time, power consumption and FLOPs. Based on these basic measures, we derive different efficiency metrics for model comparison, as discussed in Section 4.4.

4.2 Setup Details

The TextBrewer framework (Yang et al., 2020) is used for the distillation experiments, while the Huggingface Transformers[4] library is used for fine-tuning and inference. More information on hyper-parameters and data set sizes can be found in Appendix A. All experiments are executed on an Nvidia DGX-1 server with 8 Tesla V100 SXM2 32GB. In order to get measurements under realistic conditions, we run different stages in parallel on different GPUs, while blocking other processes from the system to avoid external interference. Each experimental stage is repeated 3 times and measurements of execution time and power consumption are averaged.[5]

The different cost types are measured as follows:

- **Execution time:** We average the duration of the individual Python jobs for each experimental stage.
- **Power consumption:** We measure power consumption for all 4 PSUs of the server as well as individual GPU power consumption, following Gustafsson et al. (2018). Based on snapshots of measured effect at individual points in time, we calculate the area under the curve to get the power consumption in Wh. Since we run the task-agnostic distillation using distributed data parallelism on two GPUs, we sum the consumption of both GPUs for each TA run.
- **FLOPs:** We estimate the number of FLOPs required for each stage using the estimation formulas proposed by Kaplan et al. (2020), for training (7) and inference (8):

$$\mathrm{FLOP}_T = 6 \cdot n \cdot N \cdot S \cdot B \quad (7)$$

$$\mathrm{FLOP}_I = 2 \cdot n \cdot N \cdot S \cdot B \quad (8)$$

where n is the sequence length, N is the number of model parameters, S is the number of training/inference steps, and B is the batch size. The cost for fine-tuning a model is given by FLOP_T, while the evaluation cost is FLOP_I. For distillation, we need to sum FLOP_T for the student model and FLOP_I for the teacher model (whose predictions are used to train the student model).

4.3 Basic Results

Table 1 shows basic measurements of performance and costs for different scenarios and stages. We see that the fine-tuned KB-BERT model (FT) reaches an F1 score of 87.3; task-specific distillation to a smaller model (TS) gives a score of 84.9, while fine-tuning after task-agnostic distillation (TA) only reaches 77.6 in this experiment. When comparing costs, we see that task-agnostic distillation is by far the most expensive stage. Compared to task-specific distillation, the execution time is more than 40 times longer, the power consumption almost 100 times greater, and the number of FLOPs more than 20 times greater. Although the fine-tuning costs are smaller for the distilled TA model, the reduction is only about 50% for execution time and power consumption and about 30% for FLOPs.

We also investigate whether power consumption can be predicted from the number of FLOPs, as this is a common argument in the literature for preferring the simpler FLOPs calculations over the more

[4]https://huggingface.co/docs/transformers/index

[5]Since we repeat stages 3 times for every model instance, task-specific distillation, fine-tuning of the distilled model, and evaluation of FT are repeated 9 times, while evaluation of TS and TA is repeated 27 times.

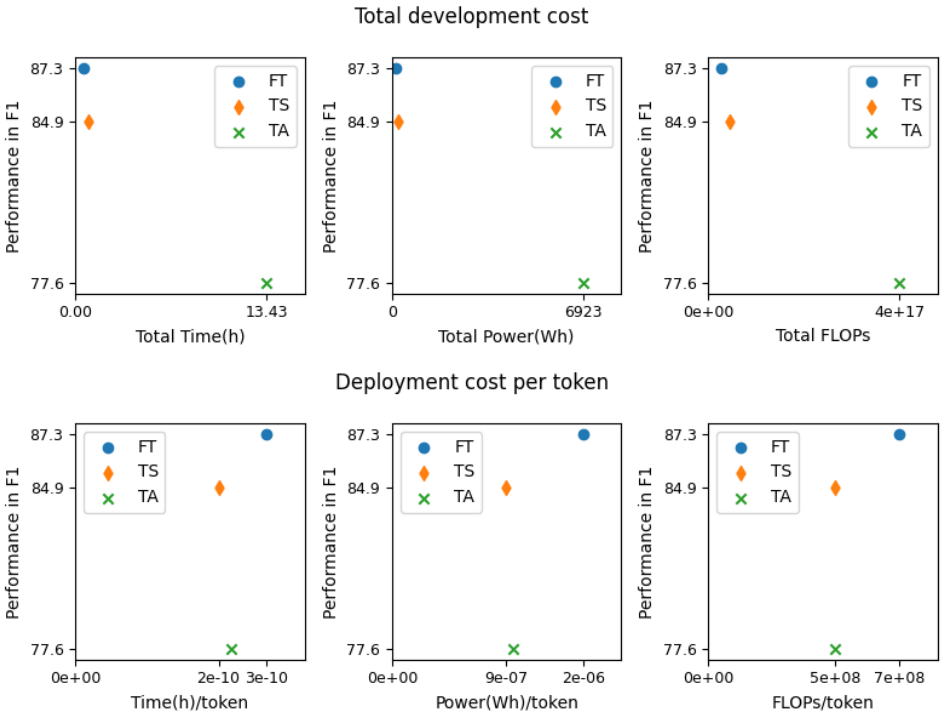


Figure 2: Pareto efficiency for the development phase (top) and the deployment phase (down) based on three different cost measures: execution time (left), power consumption (center), and FLOPs (right).

involved measurements of actual power consumption. We find an extremely strong and significant linear correlation between the two costs (Pearson $r = 0.997$, $p \approx 0$). Our experiments thus corroborate earlier claims that FLOPs is a convenient cost measure that correlates well with power consumption (Schwartz et al., 2020; Axberg, 2022). However, it is worth noting that the GPU power consumption, which is reported in Table 1 and which can thus be estimated from the FLOPs count, is only 71.7% of the total power consumption of the server including all 4 PSUs.

4.4 Measuring and Comparing Efficiency

So how do our three models compare with respect to resource-efficiency? The answer is that this depends on what concept of efficiency we apply and which part of the life-cycle we consider. Figure 2 plots product-as-performance as a function of cost separately for the development phase and the deployment phase, corresponding to Equations (3) and (4), which allows us to compare Pareto efficiency. Considering only the development phase, the FT model is clearly optimal, since it has both the highest performance and the lowest cost of all models. Considering instead the deployment phase, the FT model still has the best performance, but the other two models have lower (average) inference cost. The TA model is still suboptimal, since it gives lower performance at the same cost as the TS model.[6] However, FT and TS are both optimal with respect to Pareto efficiency, since they are both at the Pareto front given the data we have so far (meaning that neither is outperformed by a model at the same cost level nor has higher deployment cost than any model at the same performance level). In order to choose between them, we therefore have to judge whether a 2.4 point improvement in F1 score in the long run is worth the increase in execution time and power consumption, which in this case amounts to 0.077 nano-seconds and 0.607 micro-watts per token, respectively.

For a more holistic perspective on life-time efficiency, we can switch to a product-as-output model and plot deployment efficiency as a function of both the initial development cost and the average inference cost for processing new data over lifetime, corresponding to Equation (6) and our newly proposed notion of amortized efficiency. This is depicted in Figure 3, which compares the FT and

[6]It is worth noting, however, that the TA model can be fine-tuned for any number of specific tasks, which could make it competitive in a more complex scenario where we can distribute the initial distillation cost over a large number of fine-tuned models.

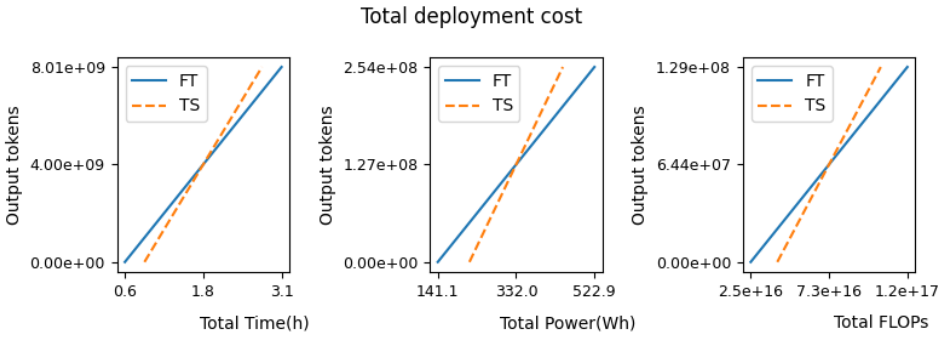


Figure 3: Amortized efficiency of the deployment phase over lifetime, based on three different cost measures: execution time (left), power consumption (center), and FLOPs (right).

TS model (disregarding the suboptimal TA model). We see that, although the FT model has an initial advantage because it has not incurred the cost for distillation, the TS model eventually catches up and becomes more time-efficient after processing about 4B tokens and more energy-efficient after processing about 127M tokens. It is however important to keep in mind that this comparison does not take performance into account, so we again need to decide what increase in cost we are willing to pay for a given improvement in performance, although the increase in this case is sensitive to the expected lifetime of the models. Alternatively, as mentioned earlier, we could weight the output by performance level, which in this case would mean that the TS model would take longer to catch up with the FT model.

Needless to say, it is often hard to estimate in advance how long a model will be in use after it has been deployed, and many models explored in a research context may never be deployed at all (over and above the evaluation phase). In this sense, the notion of life-time efficiency admittedly often remains hypothetical. However, with the increasing deployment of NLP models in real applications, we believe that this perspective on resource-efficiency will become more important.

5 Conclusion

In this paper, we have discussed the concept of resource-efficiency in NLP, arguing that it cannot be reduced to a single definition and that we need a richer conceptual framework to reason about different aspects of efficiency. As a complement to the established notion of Pareto efficiency, which separates development and deployment under a product-as-performance model, we have proposed the notion of amortized efficiency, which enables a life-cycle analysis including both development and deployment under a product-as-output model. We have illustrated both notions in a simple case study, which we hope can serve as inspiration for further discussions of resource-efficiency in NLP. Future work should investigate more sophisticated ways of incorporating performance level into the notion of amortized efficiency.

Acknowledgments

We would like to thank Jonas Gustafsson and Stefan Alatalo from the ICE data center at RISE for their help with the experimental setup of the case study. Our sincere gratitude goes also to Petter Kyösti and Amaru Cuba Gyllensten for their insightful comments during the development of this work. Finally, we wish to thank the conference reviewers for their constructive feedback.

References

Tom Axberg. 2022. Deriving a natural language processing inference cost model with greenhouse gas accounting: Towards a sustainable usage of machine learning. Master's thesis, KTH Royal Institue of Technology.

Mostafa Dehghani, Anurag Arnab, Lucas Beyer, Ashish Vaswani, and Yi Tay. 2022. The efficiency misnomer. In *Proceedings of the Tenth International Conference on Learning Representations (ICLR)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jonas Gustafsson, Sebastian Fredriksson, Magnus Nilsson-Mäki, Daniel Olsson, Jeffrey Sarkinen, Henrik Niska, Nicolas Seyvet, Tor Björn Minde, and Jonathan Summers. 2018. A demonstration of monitoring and measuring data centers for energy efficiency using opensource tools. In *Proceedings of the Ninth International Conference on Future Energy Systems*, pages 506–512.

Daniel Hershcovich, Nicolas Webersinke, Mathias Kraus, Julia Anna Bingler, and Markus Leippold. 2022. Towards climate awareness in nlp research. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2480–2494.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. arxiv:2011.08361.

Xiangyang Liu, Tianxiang Sun, Junliang He, Jiawen Wu, Lingling Wu, Xinyu Zhang, Hao Jiang, Zhao Cao, Xuanjing Huang, and Xipeng Qiu. 2022. Towards efficient NLP: A standard evaluation and a strong baseline. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3288–3303.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the National Library of Sweden – Making a Swedish BERT. arXiv:2007.01658.

Carl Mitcham. 1994. *Thinking through Technology: The Path between Engineering and Philosophy*. The University of Chicago Press.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Communications of the ACM*, 63(12):54–63.

Språkbanken. 2022. SUCX 3.0: Stockholm-Umeå corpus 3.0 scrambled. https://spraakbanken.gu.se/en/resources/sucx3.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650.

Robert Endre Tarjan. 1985. Amortized computational complexity. *SIAM Journal on Algebraic and Discrete Methods*, 6(2):306–318.

Marcos Treviso, Tianchu Ji, Ji-Ung Lee, Betty van Aken, Qingqing Cao, Manuel R. Ciosici, Michael Hassid, Kenneth Heafield, Sara Hooker, Pedro H. Martins, André F. T. Martins, Peter Milder, Colin Raffel, Edwin Simpson, Noam Slonim, Niranjan Balasubramanian, Leon Derczynski, and Roy Schwartz. 2022. Efficient methods for natural language processing: A survey. arXiv:2209.00099.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Ziqing Yang, Yiming Cui, Zhipeng Chen, Wanxiang Che, Ting Liu, Shijin Wang, and Guoping Hu. 2020. TextBrewer: An Open-Source Knowledge Distillation Toolkit for Natural Language Processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 9–16.

A Experimental Details

A.1 Data Sets

The SUCX 3.0 dataset (simple_lower_mix version)[7] is used for fine-tuning, task-specific distillation and evaluation. The dataset splits are are the following: 43126 examples in the training set, 10772 in the validation set and 13504 examples in the test set.

For task-agnostic distillation, we are using a deduplicated version of Swedish Wikipedia, with the following dataset split: $2,552,479$ sentences in the training set and $25,783$ sentences in the validation set.

A.2 Models and Hyperparameters

The base model in our experiments is KB-BERT-cased.[8] The hyperparameters used for fine-tuning and distillation are presented in Table 2. In the fine-tuning experiments, early stopping is used and the best performing model in the validation set is saved. The task-agnostic distillation experiments are performed on two GPUs, using the distributed

[7] https://huggingface.co/datasets/KBLab/sucx3_ner
[8] https://huggingface.co/KB/bert-base-swedish-cased

data parallel functionality of pytorch, while gradient accumulation steps are set to 2.

	FT	TS	TA	Eval
Batch size	32	32	8	32
Training epochs	10	2	0.75	–
Sequence length	256	256	256	256
Learning rate	0.00003	0.00005	0.0001	–
Warm-up steps	404	260	3750	–

Table 2: Hyperparameters for FT, TS, TA and Eval.

Identifying Token-Level Dialectal Features in Social Media

Jeremy Barnes[1], **Samia Touileb**[2], **Petter Mæhlum**[3], **Pierre Lison**[4]

[1]University of the Basque Country, [2]University of Bergen,
[3]University of Oslo, [4]Norwegian Computing Center
jeremy.barnes@ehu.eus, samia.touileb@uib.no,
pettemae@ifi.uio.no, plison@nr.no

Abstract

Dialectal variation is present in many human languages and is attracting a growing interest in NLP. Most previous work concentrated on either classifying dialectal varieties at the document or sentence level or performing standard NLP tasks on dialectal data. In this paper, we propose the novel task of *token-level dialectal feature prediction*. We present a set of fine-grained annotation guidelines for Norwegian dialects, expand a corpus of dialectal tweets, and manually annotate them using the introduced guidelines. Furthermore, to evaluate the learnability of our task, we conduct labelling experiments using a collection of baselines, weakly supervised and supervised sequence labelling models. The obtained results show that, despite the difficulty of the task and the scarcity of training data, many dialectal features can be predicted with reasonably high accuracy.

1 Introduction

Language variation is a pervasive phenomenon in human language. These varieties can differ on phonemic, lexical, or syntactic levels, among others, and often vary on several levels at a time (Chambers and Trudgill, 1998). One common type of language variation stems from geographical location, as people actively use regional variations to mark their identity. When a language variety indicates *where* a speaker is from, we call this variety a **dialect**, or more precisely a **geolect** or **topolect**, as the word 'dialect' can also refer to social background or occupation. In this work, we use 'dialect' to denote geographical variation.

Dialectal variation in Norwegian is widespread and, in contrast to many languages, the use of spoken and written dialects in the public sphere is generally viewed positively (Bull et al., 2018). Although Norwegian can be broadly divided into four dialectal regions, many dialectal features are shared across these regions (see Figure 1). Therefore, rather than seeing dialects as discrete categories, we should view them as a combination of correlated dialectal features (Nerbonne, 2009).

The under-resourced status of dialects, however, makes it difficult to build NLP tools from scratch. This is exacerbated by the growing reliance on pre-trained language models, which often encounter few examples of dialectal data during training. If NLP models fail to process dialectal inputs, their deployment may reinforce existing inequalities, as those who use a non-standard variety will either receive worse service or be forced to adopt a standard variety to interact. Those who advocate for maintaining dialectal variation also depend on tools to help them monitor the use of dialects on social media. This motivates the development of fine-grained models of dialectal features.

Previous work on dialectal NLP has classified dialects, geographical location, or provided training and testing resources for various dialects. In this paper, we take a different viewpoint on identifying dialects, opting to label the *token-level dialectal features* of a text rather than classifying or predicting the geolocation of the entire text. We first propose a fine-grained annotation scheme for token-level dialectal features in Norwegian. We then annotate a corpus of Norwegian dialectal tweets using this scheme, and finally validate its use for fine-tuning neural sequence labeling models in Norwegian.

Our contributions are 1) we introduce the novel task of **token-level dialect feature identification**, 2) provide a **novel corpus of Norwegian dialectal tweets** annotated for 21 token-level features,[1] and 3) describe **extensive experiments** demonstrating the learnability and difficulty of the task.

[1]Annotation guidelines, procedure and data available at https://github.com/jerbarnes/nordial

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 146–158

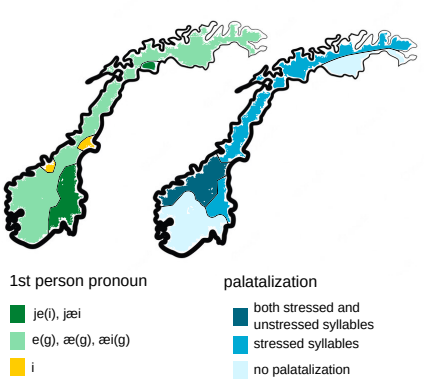


Figure 1: Map of two dialectal features in Norwegian that do not coincide geographically.

2 Related Work

In contrast to more formal writing, social media abounds with dialectal variation, ranging from variation between racial groups (Eisenstein, 2015), to variation within online communities (Danescu-Niculescu-Mizil et al., 2013). While not all levels of variation are equally present, often due to a speaker's lack of awareness of sociolinguistic indicators (Labov, 2006), a substantial share of dialectal variation is reliably transcribed in social media posts (Eisenstein, 2013; Doyle, 2014).

For NLP, dialectal data presents both a challenge and an important area to improve upon. Previous work in NLP has included descriptive corpus studies (Jones, 2015; Tatman, 2016), dialect classification (Zampieri et al., 2017), geolocation of tweets based on their dialectal features (Eisenstein et al., 2010; Hovy and Purschke, 2018) or quantifying the spatial dependence of linguistic variables (Nguyen and Eisenstein, 2017).

There have also been a series of workshops (VarDial) (*e.g.* Nakov et al., 2016, 2017; Zampieri et al., 2018) that include work on discriminating similar languages (Haas and Derczynski, 2021), identifying dialects (Jauhiainen et al., 2021), and geolocation of tweets (Gamăn et al., 2021). The workshops have also held several shared tasks with the aim to identify languages and dialects (Zampieri et al., 2017), as well as morpho-syntactic tagging (Zampieri et al., 2018). Another series of shared tasks have focused on the identification of Arabic dialects (Bouamor et al., 2019; Abdul-Mageed et al., 2020, 2021). While each of these shared tasks proposed dialect identifications on different level of granularity (region, country, and city-levels), they all approached dialect identification as a sentence classification task. Work on code-switching (*e.g.* Solorio and Liu, 2008; Jain and Bhat, 2014; Samih et al., 2016; Çetinoğlu, 2016), on the other hand, has focused on word-level classification, but usually casts this a binary decision, rather than identifying fine-grained labels.

Regarding Norwegian dialects specifically, linguistic work is long and varied. Christiansen (1954) described the main dialect regions, while Sandøy (2000) describes several factors that drive language change in modern Norwegian dialects, *e.g.* urban jumping (Chambers and Trudgill, 1998), the prestige of certain dialects, or the general tendency towards simplification. Within NLP, Barnes et al. (2021) present the *NorDial* corpus, a curated collection of 1,073 tweets classified as either Bokmål, Nynorsk (which are the two standardized written forms for Norwegian), dialectal, or mixed. The authors experimented with classifying these tweets with Norwegian BERT models (Kummervold et al., 2021) and found that the resulting models achieved reasonably good performance at identifying tweets written in dialectal Norwegian.

Demszky et al. (2021) introduce the task of dialect feature detection at the phrase/sentence level. They use available annotations on the ICE-India English data (Greenbaum and Nelson, 1996) and annotate a small amount of this data with separate set of 18 dialectal features. As they have no training data for their annotated features, they propose to use a minimal pairs framework as a kind of weak supervision. They find that even with minimal supervision, their models are able to reliably predict many of the features. However, they do not predict which tokens carry the features, choosing to label the entire phrase instead.

To address these limitations, we propose a new approach where we annotate dialectal features at the token level. We contend that this annotation strategy provides a more fine-grained view of the actual use of dialectal features in social media.

3 Dialectal tweet collection

In order to increase the number of dialectal tweets, we expand upon the NorDial corpus (Barnes et al., 2021) and collect a further 3,000 tweets to annotate. During the initial collection, we used the Twitter API without a search query and confined the search to tweets from the geographical area of Norway. This first collection, however, yielded relatively

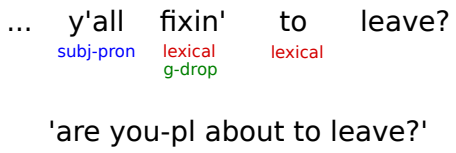


Figure 2: Example of Texan English with dialectal labels below each token.

few dialectal tweets and those found displayed a narrow set of dialectal features. To increase the variety of dialectal features, we first collected a list of dialectal features from the *Store Norske Leksikon*[2] (Norwegian Encyclopedia) and used these as queries in the Twitter API. We then identified users whose tweets often contain these dialectal features and collected their tweets, as well as tweets from their followers. As many of the collected tweets were still written in standard Bokmål or Nynorsk, three annotators were asked to classify the tweets, and those labelled as dialectal were then included in the process of fine-grained feature annotation. In total, 2,455 of 3000 tweets were classified as dialectal.

4 Annotation of fine-grained dialectal features

Figure 2 shows an example from Texan English with three main dialectal features: *y'all*, which is the non-standard second person plural pronoun and *fixin' to*, which contains the lexical feature 'fixing to' which means 'about to', and the morphological feature of 'g-dropping'.

In the rest of this section, we detail the inventory of dialectal features used in our annotation. As each example highlights a minimal pair example of a single dialectal feature, we do not include the labels below the relevant tokens.

4.1 Dialectal features

The inventory of dialectal features stems from the linguistic traits that can be encountered in written form as described by Venås and Skjekkeland (2022). Other dialectal features, such as differing toneme patterns or the pronunciation of 'l', were not considered, as they are not observable in written texts. We focus on the dialectal impact a word has, *i.e.* whether the annotator can determine that the word falls outside of the norms in such a way as to identify the speaker as a dialect user. For example, a form like *jæ* for 'I' has a higher impact than the choice between the two habitual aspect markers *bruke* and *pleie*, 'use (to)', as the latter are both part of the written norm, and the former is unlikely to be an accidental misspelling.

In cases where there are several choices of form, some of these might be more marked than others. In the following examples, we show the original dialectal version and normative Bokmål versions: **dialect/normative** and the English translation.

Subject and object pronoun use Pronouns are extremely common dialect markers in Norwegian, as a single pronoun can be marked enough to identify the dialect of the writer. We label the subject and object (or oblique) functions separately, but do not include a separate label for the dative.

(1) ... og **dem/de** blir aldrig eldre ...
'... and they never get older ...'

Copula The copula 'være/vera/vere' (be) is marked with the label copula. We only mark dialectally interesting, non-standard versions of the copula, such as 'e' and 'værra'.

(2) Det **e/er** rart at ...
'It is weird that ...'

Contraction We label contractions for negation adverb 'ikke/ikkje' (not), and enclitic pronouns. The verb and the adverb are labeled separately, but both are labeled with the *contraction* label.

(3) **ekke/er ikke** han som skulle ...
'he is not the one who should have' ...

Palatalization In Norwegian palatalization occurs frequently to geminated consonants such as 'nn', 'dd' and 'll', in several dialects. In writing it is usually indicated by additions of 'j' or 'i'.

(4) ho e nok **forbainna/forbanna** ...
'She is so angry ...

Present marker deletion In some dialects the final '-r' that marks the present tense for many verbs in both Bokmål and Nynorsk is dropped. We also use this label to indicate the dropping of '-l' in present tense verb forms such as 'skal' → 'ska' (will) and 'vil' → 'vi' (want).

[2] https://snl.no/

Apocope Apocope is the loss of word-final '-a' or '-e' and is common in certain dialects.

(5) Æ e her for å **vinn/vinne**
'I am here to win' ...

Voicing Voicing is the process by which consonants which are voiceless in some dialects become voiced, where 'p', 't', and 'k' become 'b', 'd', and 'g', respectively.

(6) Eg kommer ikkje **tebage/tilbake**
'I won't come back' ...

Vowel shift Both monophtongal changes such as lowering (e→æ) and dipthongization such as 'e' → 'ei' are all marked with the vowel shift label. We also see cases of monophthongization such as 'ei' → 'ø'. One important heuristic we follow is that we do not mark vowel shift in words that are tagged with any of the pronoun labels.

Lexical variation This label is used when the lemma of a word is notably marked. Loanwords are not affected by this; the word has to be a dialectal or local version of a standard word that could have been used instead. An example is the word 'tue' (towel) instead of 'klut' (cloth).

Demonstrative pronoun use In some dialects it is common to use third-person pronouns as determiners together with proper names. These can be full forms as in 'ho Kari' (she Kari) or 'han Olav' (he Olav) or reduced as in 'a Kari' or 'n Olav'.

Shortening In some dialects, writers indicate a change of stress to the first syllable with accompanying vowel reduction and consonant lengthening, by writing a double consonant after the first syllable if there is originally only one, as in 'pottet' instead of 'potet' (potato).

Grammatical gender of nouns The grammatical gender of nouns in Norwegian has considerable variation. The least common remnant of the feminine gender is the indefinite article 'ei'. Keeping the feminine definite form '-a' is more common, but there is also a clear tendency to see certain high-frequency words as feminine. Examples are words like 'jente' (girl). 'Ei jente' (a girl) is slightly marked towards favoring the feminine form, while 'jenten' (the girl) is strongly marked towards a dialect with no feminine gender.

Marked This label is used for words that are part of the written languages' norms, but which are still rarely used, and therefore dialectally marked. An example is the question word 'åssen' (how), which is accepted in Bokmål, but still infrequent, and somewhat marked compared to 'hvordan' (how).

h-v A notable difference between Bokmål and Nynorsk is that Nynorsk has 'kv' where bokmål has 'hv', especially for interrogatives. In some dialects, the 'v' is lost, giving only 'k' or 'h', as in *'hårr'* for 'hvor' (where) or *'ka'* for 'hva' (what). This is marked with the *h-v* label. Any token with this label will not have the *phonemic spelling* label.

Adjectival declension This labels is used for adjectives with non-standard endings, such as '-e' in indefinite or non-plural environments.

(7) ein **gode/god** venn
'a good friend' ...

Nominal declension This label is used when a noun takes a non-standard declensional ending.

(8) Fortsatt gode **muligheta/muligheter** til gå
'still good chances to go' ...

Verb conjugation This label is used when a verb takes a non-standard conjugation ending, such as 'skrivi' for 'skrive' (to write).

Functional words The dialectal forms of many functional words are spelled radically different. We label all functional words whose spellings are not in accordance with the written norms.

(9) Tru **ittæ/ikke** dæ æ dær
'do not think it is there' ...

Phonemic spelling In cases where there is no clear dialectal variation, but it is clear that the speaker wants to indicate that they are writing a more oral form, the label phonemic spelling is used. This is mostly for cases where a pronunciation is close to the perceived norm of some standard, like 'næi' for 'nei' (no).

Interjection This label is used for all interjections, dialectal or not, such as the greeting 'heia' (hey).

4.2 Annotation procedure

For the token-level annotations, we take the tweets that were classified as dialectal in the first round, combined with the dialectal tweets from Nordial (Barnes et al., 2021). The annotation was performed by three hired student research assistants with a background in linguistics and with Norwegian as native language. All annotators are from eastern Norway, and native speakers of the eastern dialect. The first 50 tweets were annotated independently by two annotators. This first round provided the basis for group discussions, held regularly during the first phase of annotation, after which the guidelines were updated. The doubly annotated documents were then adjudicated by a third annotator after a final round of discussions concerning difficult cases. Annotators had the possibility to discuss any potential problems during both the annotation and adjudication period, but were encouraged to follow the guidelines as strictly as possible. The annotation and adjudication were both performed using the web-based annotation tool Brat (Stenetorp et al., 2012).

4.3 Annotation results and statistics

Table 1 presents the statistics for the final annotated data. We create separate test and developments splits of 500 and 300 tweets respectively, maintaining the overall distribution of labels evenly throughout the splits and leave the remaining 1,655 tweets as training data. The average length of the tweets is around 25 tokens, with an average of 4.5 annotations per tweet. Most tokens in a tweet are not annotated (84.3%), leaving an average of 0.2 annotations per token. Of the remaining 15.7%, the average number of labels per token is 1.2. In other words, 14% (1343 tokens) of the annotated tokens have multiple labels, while the remaining 86% (8167 tokens) have a single label.

Figure 3 shows the distribution of the annotated labels. Vowel shift is the most common label, followed by subject pronoun, and functional. This is expected as vowel shift covers a large number of phenomena, and subject pronoun and functional are highly salient features in Norwegian dialects. The least common are interjection, demonstrative pronoun, and gender. While these features may be more common in spoken dialects, it seems writers of tweets use them less frequently, possibly because they are much more marked when written. See the Appendix for further analysis.

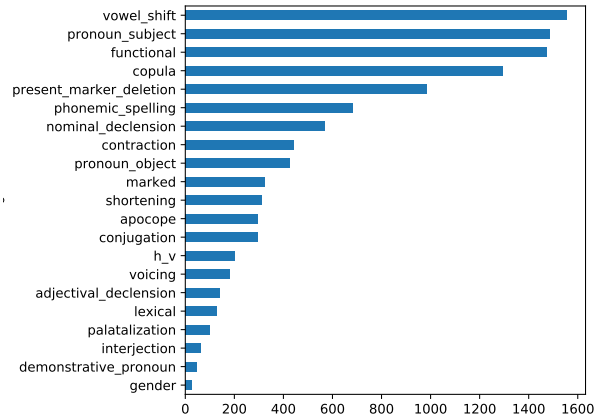


Figure 3: Frequency counts of dialectal features annotated in the full dataset of Norwegian tweets.

After completing the annotation process, the annotators pointed out that some dialectal areas (especially the Trøndersk-Central dialect) seem to be more common in the data. This might skew the label distribution to a degree.

4.4 Inter-annotator agreement

Chance-corrected inter-annotator agreement is important to determine the reliability of annotated data. The annotation we propose requires *unitization* or delimiting spans of words, *categorization*, and is inherently *multi-label*. Typical inter-annotator agreement measures, *e.g.* κ (kappa) (Cohen, 1960) or α (alpha) (Krippendorff, 1980), do not provide a good statistical basis for determining agreement with multi-labels which can span several tokens. We therefore use the γ (gamma) agreement from Mathet et al. (2015) instead, which allows for chance corrected agreement between annotators given the three above requirements.

γ combines alignment and comparing of categorization into a single chance-corrected metric. It first selects the alignment that leads to the least overall disagreement γ_o and then calculates the expected disagreement γ_e by sampling from the existing annotations. Finally, as with other measures based on disagreement, gamma is calculated as $\gamma = 1 - \frac{\gamma_o}{\gamma_e}$, where the observed measure is divided by the expected measure. Values in gamma range from $-\infty$ to 1, where 0 represents chance agreement. We use the `pygamma-agreement` package (Titeux and Riad, 2021) available in python.

The double annotations from the first and second round achieve $\gamma = 0.63$, and $\gamma = 0.64$ respectively, which we take to indicate good agreement, given

	train	dev	test	total
number of tweets	1,655	300	500	2,455
number of tokens	40,483	7,563	12,597	60,643
average number of tokens per tweet	24.5	25.2	25.2	24.7
average number of annotations per tweet	4.5	4.4	4.5	4.5
average number of annotations per token	0.2	0.2	0.2	0.2
average number of labels per annotated token	1.2	1.2	1.2	1.2

Table 1: Statistics of the dialectal feature annotations.

that the task is challenging. Common disagreements between annotators include whether a token should be considered functional or not, the use of the lexical label and the identification of vowel shift.

5 Experiments

We now describe the experimental setup employed to validate our annotations. As early results indicated that standard models had difficulty learning multi-label sequence labelling tasks, we merge occurrences of multiple labels, yielding a total of 159 combinations (including 'Ø', the null label). For each possible combination of labels in our dataset, we create a new merged label that represents them. This increases the number of total classes to be predicted, but reduces the task to a much simplified multi-label sequence labelling problem. Formally, the task is then given a sequence of N tokens $S = \{t_1, t_2, \ldots, t_n\}$ to predict the sequence of token-wise labels $L = \{l_1, l_2, \ldots, l_n\}$, where these labels can be either single labels, *e.g.*, 'vowel_shift' or a merged label, *e.g.*, 'lexical-vowel_shift'. For all experiments with neural models, we train an set of five models with different random seeds and report both micro-averaged F_1 and standard deviation.

5.1 Initial baseline

The first baseline consists of a simple majority voter that always predicts the most common label, which is 'vowel shift'.

5.2 Handcrafted functions

To investigate the extent to which the dialectal features can be inferred from known linguistic rules, we designed a set of handcrafted functions. One team member with a linguistics background and access to the annotation guidelines and the labelled training data implemented a set of 39 programmatic labeling functions, divided in three groups:

Heuristic functions Many labels can be detected programmatically. For example, to identify dialectal demonstrative pronouns, we create a function that detects demonstrative pronouns occurring within two tokens after a proper name.

Lexicon functions: Categories such as *h-v*, *functional*, or *interjection* correspond to (roughly) closed classes which can be directly compiled in lexicons. We also construct lexicons for other categories such as *marked* or *phonemic spelling*, although those categories are more productive and are not restricted to a closed set. Those lexicons are created by enumerating tokens associated with the corresponding tag in the development set.

Dictionary-based functions We can also predict a *voicing* tag by changing a soft consonant ('b', 'd', 'g') to its hard consonant ('p', 't', 'k') and then performing a lookup in precompiled dictionaries for Bokmål and Nynorsk[3].

The results of all labelling functions can then be aggregated into a unified prediction over possible labels. This aggregation is done using a Hidden Markov Model (HMM) or a majority voter (MV), as implemented in `skweak` (Lison et al., 2021).

5.3 Weakly supervised models

Handcrafted functions remain hampered by their limited coverage and lack of robustness to noise. *Weak supervision* can partially alleviate those limitations. Weak supervision operates by defining labeling functions and applying those on large amounts of unlabeled data to create a silver corpus, which is in turn employed to train a machine learning model for the task. We use the same 39 labelling functions as above and apply them to a set of 2,169 additional dialectal tweets collected similarly to the training data. Note that this data was not annotated by hand and serves mainly as a way to increase the size of the silver data with the hope of increasing recall. The outputs of those

[3]We rely here on the Norsk Ordbank for both Bokmål (`https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-5/`) and Nynorsk (`https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-41/`) and extract all inflected forms from those.

Model	Dev		Test	
'Vowel shift'	3.7		4.4	
Labeling functions (MV-aggregated)	15.6		16.4	
NB-BERT fine-tuned on HMM-aggregated weak labels	14.1	± 0.3	21.2	± 0.7
NB-BERT fine-tuned on MV-aggregated weak labels	29.7	± 0.6	33.3	± 0.7
SVM + NB-BERT embeddings (gold labels)	45.5		47.7	
BiLSTM fine-tuned on train (gold labels)	38.5	± 3.4	45.5	± 0.0
NorBERT fine-tuned on train (gold labels)	42.0	± 6.0	52.9	± 1.3
NB-BERT fine-tuned on train (gold labels)	54.9	± 0.8	58.4	± 0.4

Table 2: Micro F_1 on dev and test for the vowel shift baseline, handcrafted labelling functions, weakly supervised models aggregated with either Hidden Markov Models (HMM) or majority voting (MV), and supervised models (BiLSTM, NorBERT, NB-BERT) trained on gold labels from the training set. The results for neural models are shown as the average and standard deviation of five runs with different random seeds.

functions are then aggregated using either HMMs or majority voting. After aggregation, we train an NB-BERT (Kummervold et al., 2021) model on this silver data using the same procedure as the supervised models described in the next section.

5.4 Supervised models

We test one context-free model and three sequence labeling models which take context into account: a bidirectional LSTM and two Norwegian pretrained language models. Those models are all fine-tuned on the gold labels of the training set.

The context-free model is an linear SVM trained using the embeddings from the NB-BERT model (see below). Specifically, we create vector representations for each word in the training data by passing the words individually to the embedding layer of NB-BERT. For words that are split into several subcomponents due to the byte pair tokenization, we take the average representation of these embeddings. Finally, we train a linear SVM classifier[4] and fine tune the C parameter on the dev set. This model therefore uses the same representation strategy as the stronger NB-BERT model, but uses these without contextualization and has significantly fewer trainable parameters.

The BiLSTM is a two layer Bidirectional LSTM (Schuster and Paliwal, 1997) with 100-dimensional pre-trained embeddings,[5] and a hidden layer size of 256. The embeddings were trained on the Norwegian Newspaper corpus, the Norwegian Web as corpus (NoWaC) (Guevara, 2010), and NBDigital corpus (books from the national library of Norway), using fastText Skipgram (Bojanowski et al., 2017), and with a vocabulary size of 4,428,648 tokens. We train the BiLSTM model for a maximum of 50 epochs with a patience of 3 using Adam (Kingma and Ba, 2014) with default parameters.

The transformer models include NorBERT (Kutuzov et al., 2021) and NB-BERT (Kummervold et al., 2021). NorBERT is a BERT (Devlin et al., 2019) model trained from scratch, including the subword tokenizer, on the Norwegian Newspaper corpus combined with Wikipedia dumps for Bokmål and Nynorsk, for a total of nearly 2 billion tokens. The NB-BERT model is a multilingual BERT base model further trained on the Norwegian Colossal Corpus.[6] The latter is therefore less adapted to Norwegian vocabulary, but has been exposed to a larger volume and variety of Norwegian texts, including dialectal context.

As commonly done, we add a classification head to the transformer models and rely on the Huggingface library (Wolf et al., 2020) for the implementation. To deal with subword tokens, we assign the token label only to the first subword and mask the others. We use a learning rate of 2e-5, a weight decay of 0.01, and a batch size of 16 with Adam W (Loshchilov and Hutter, 2019). We train the models for 20 epochs, updating both pretrained weights and classification heads, and do not tune any parameters on the development set.

6 Results

Table 2 shows the micro-average F_1 scores obtained by all approaches on the test set.

The majority label baseline ('vowel shift')

[4] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[5] Model 81 downloaded from the NLPL word embedding repository http://vectors.nlpl.eu/repository/

[6] https://github.com/NbAiLab/notram/blob/master/guides/corpus_description.md

achieves a low F_1 score of 4.4. While the handcrafted functions obtain slightly higher F_1 scores than these baselines, the scores demonstrate that the proposed task is challenging and that simple rule-based approaches are insufficient.

All supervised models perform better than the weak supervision models, with the BiLSTM achieving 45.5 F_1, the SVM 47.7, NorBERT 52.9, and NB-BERT 58.9. In general, the results of the SVM follow a high-precision low-recall pattern (*e.g.*, h-v: precision 90/recall 42, interjection: 100/8.3, palatalization: 100/13.3) displaying this model's inability to generalize to new examples, while the neural models tend to generalize better. The good performance of NB-BERT follows previous trends for classification of tweets (Barnes et al., 2021). Those results differs from Demszky et al. (2021), who found that the weak supervision provided by several hundred minimal pairs was often enough to outperform supervised approaches. This discrepancy may be due to differences in the training set size or the increased difficulty of labeling the tokens rather than the full utterance.

Label	Precision	Recall	F_1
copula	**94.5**	**94.8**	**94.7**
pron. subj.	**82.9**	**74.3**	**78.4**
pm deletion	**72.4**	**79.9**	**76.0**
pron. obj.	**88.2**	63.8	**74.0**
h-v	67.4	69.0	68.2
functional	**71.2**	63.9	67.3
voicing	**73.7**	58.3	65.1
apocope	**75.5**	53.6	62.7
nom. decl.	66.0	55.6	60.4
dem. pro.	60.0	60.0	60.0
contraction	**77.1**	45.8	57.4
vowel shift	58.4	55.3	56.8
phon. spelling	40.7	36.5	38.5
shortening	41.3	35.2	38.0
adj. decl.	36.8	28.0	31.8
palatalization	**75.0**	20.0	31.6
interjection	30.0	25.0	27.3
conjugation	24.3	15.8	19.1
marked	6.7	8.0	7.3
lexical	50.0	3.0	5.7
gender	0.0	0.0	0.0

Table 3: Precision, recall, and F_1 scores of NB-BERT.

7 Error Analysis

We provide here an error analysis of the results from the best performing model, namely NB-BERT. Table 3 shows the per-label precision, recall, and F_1 scores of the NB-BERT model. We highlight scores > 70 in blue and scores < 50 in red. The model performs well on *copula*, *pronouns* (subject and object), and *present marker deletion*. It performs poorly on *phonemic spelling*, *shortening*, *adjectival declension*, *interjection*, *conjugation*, *marked*, *lexical*, and *gender*. There is a statistically significant correlation between frequency in the training corpus and F_1 (Spearman's $\rho = 0.65$, $p = 0.001$), although there are outliers such as *vowel shift*. This may be due to the range of heterogeneous contexts in which *vowel shift* can occur. Other labels such as *functional* or *h-v* are more difficult than expected, likely due to the number of possible forms.

It is clear from the confusion matrix in Figure 4 that the model confuses most labels with the label 'Ø'. The other label that is regularly over-predicted is 'vowel shift', which suggests that frequency plays a strong role in prediction. When it comes to multiple labels, the performance of NB-BERT can be characterized as high-precision and low-recall, with only 30% of the test tokens with multiple labels being predicted as such by the model, with a micro F_1 of 88.1.

To establish the importance of context, we compare the performance of the SVM and NB-BERT models on context-free labels (h-v, functional, vowel shift, voicing, palatalization, shortening, interjection, nominal declension, conjugation, marked, lexical) and context-sensitive labels (phonemic spelling, contraction, pronoun subject, pronoun object, present marker deletion, apocope, adjectival declension, demonstrative pronoun, copula, gender). We compare these two groups by taking the average difference between the F1 scores for each label. For the context-free labels, there is an average 14.0 percentage points difference between the two models, while for the context-sensitive labels, this difference is 24.9. This implies that including context via contextual embeddings is especially important for the context-sensitive labels.

8 Conclusion and future work

In this paper, we have presented a new dataset for token-level dialect feature prediction, composed of Norwegian tweets classified as dialectal, which we annotated for 21 dialectal features achieving good

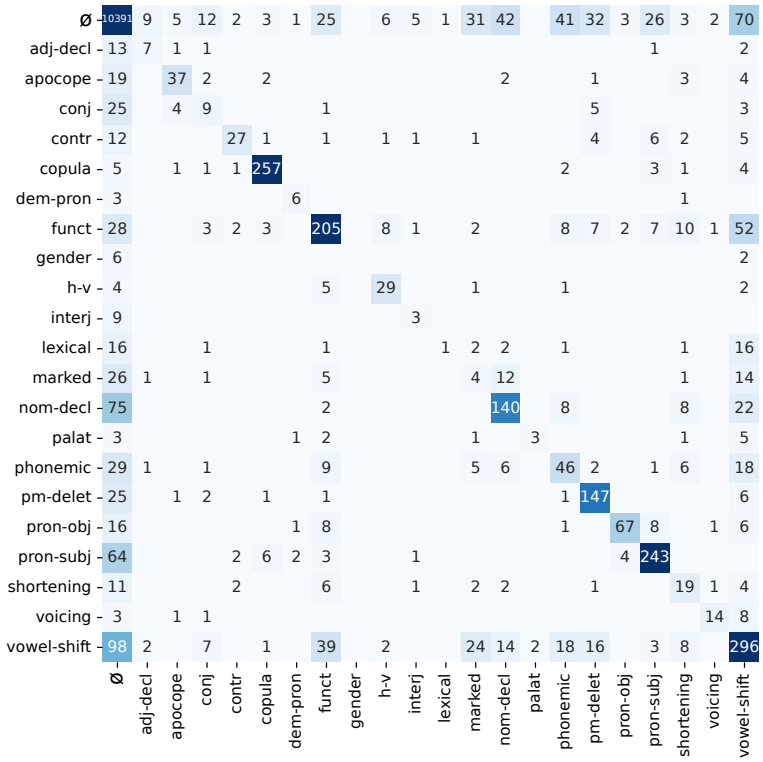


Figure 4: Confusion matrix of the NB-BERT model. 'Ø' represents predicting no label. The y-axis represents the true labels.

inter-annotator agreement. This dataset was employed in a set of labelling experiments including rule-based approaches, weakly supervised, and supervised neural models. The experimental results corroborate the difficulty of the task, with micro F_1 scores ranging from 16.4 for handcrafted functions to 58.9 for the best supervised model.

This work provides a basis for future research on dialectal features. Specifically, we plan to explore the distribution of these dialectal features in different online communities using the learned models. The data can also help multi-task learning of text normalization models, as identifying tokens to be normalized should lead to improvements.

Another promising direction is to predict regional dialects based on the token-level features. As dialectal traits are correlated with certain regions, it may be possible to create hierarchical representations of dialects on different levels of granularity. The guidelines, models, and annotations will be made publicly available.[7]

As for potential risks, the dataset was compiled from social media posts. Therefore, complying with the GDPR regulations, authors of these posts must have the right to be forgotten if they wish to remove previous posts. We will therefore only release the annotations with the original tweets upon request. In this way, if they have been deleted, they will also not be recuperated for our dataset.

Acknowledgements

This work has been partially supported by the Tekstub initiative at the University of Oslo, MediaFutures, the HiTZ center and the Basque Government (Research group funding IT-1805-22).

We also acknowledge the funding from the following projects: DeepKnowledge (PID2021-127777OB-C21) project funded by MCIN/AEI/10.13039/501100011033 and by FEDER Una manera de hacer Europa.

Parts of this work was supported by industry partners and the Research Council of Norway with funding to MediaFutures, project number 309339.

Finally we want to thank the two annotators Alexandra Wittemann and Marie Emerentze Fleisje for their annotation efforts.

[7] https://github.com/jerbarnes/nordial

Appendix A – Limitations

Our motivation for this project was to take a first step towards fine-grained dialectal feature detection. However, there are several limitations with the current annotation process and modeling approaches presented in this paper.

Firstly, although the idea of identifying dialectal features in Twitter data is rather general, the guidelines and dataset provided with this paper are specific to Norwegian. While we hope that these resources are helpful to other language variations, adapting this to another situation would require a non-trivial amount of work and money. The creation of this dataset required 7000 euro.

The annotation procedure focused on token-level labels. Dialectal features that arise from the absence of a given token (*e.g.* subject dropping, as in Example 10) or that cannot be marked at the token-level (*e.g.* non-V2 word order in interrogative sentences as in Example 11) are therefore not explicitly annotated in this dataset.

(10) *Spent på det ...*
Exited on it ...
'(I am) excited about it'

(11) *Ka du sier?*
What you say?
'What are you saying?'

Appendix B – Co-occurence of annotated labels

Figure 5 shows the co-occurrence of the 21 labels at token-level. From the figure, it is clear that most labels do not co-occur or do so rarely. The labels that co-occur the most frequently are vowel shift and functional (366), vowel shift and present marker deletion (121), functional and contraction (104), phonemic and functional (65) and pronoun subject and contraction (57). Vowel shift, besides being the most common label, is also the label that co-occurs the most with other labels.

References

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020. NADI 2020: The first nuanced Arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 97–110, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Jeremy Barnes, Petter Mæhlum, and Samia Touileb. 2021. NorDial: A preliminary corpus of written Norwegian dialect use. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 445–451, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The MADAR shared task on Arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207, Florence, Italy. Association for Computational Linguistics.

Tove Bull, Espen Karlsen, Eli Raanes, and Rolf Theil. 2018. *Norsk språkhistorie*, volume 3. Novus, Oslo.

Özlem Çetinoğlu. 2016. A Turkish-German code-switching corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4215–4220, Portorož, Slovenia. European Language Resources Association (ELRA).

Jack K. Chambers and Peter Trudgill. 1998. *Dialectology*, 2 edition. Cambridge Textbooks in Linguistics. Cambridge University Press.

Hallfrid Christiansen. 1954. *Hovedinndelingen av norske dialekter*, volume 1954. Bymålslaget Oslo.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37.

Cristian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. No country for old members: user lifecycle and linguistic change in online communities. *Proceedings of the 22nd international conference on World Wide Web*.

Dorottya Demszky, Devyani Sharma, Jonathan Clark, Vinodkumar Prabhakaran, and Jacob Eisenstein. 2021. Learning to recognize dialect features. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2315–2338, Online. Association for Computational Linguistics.

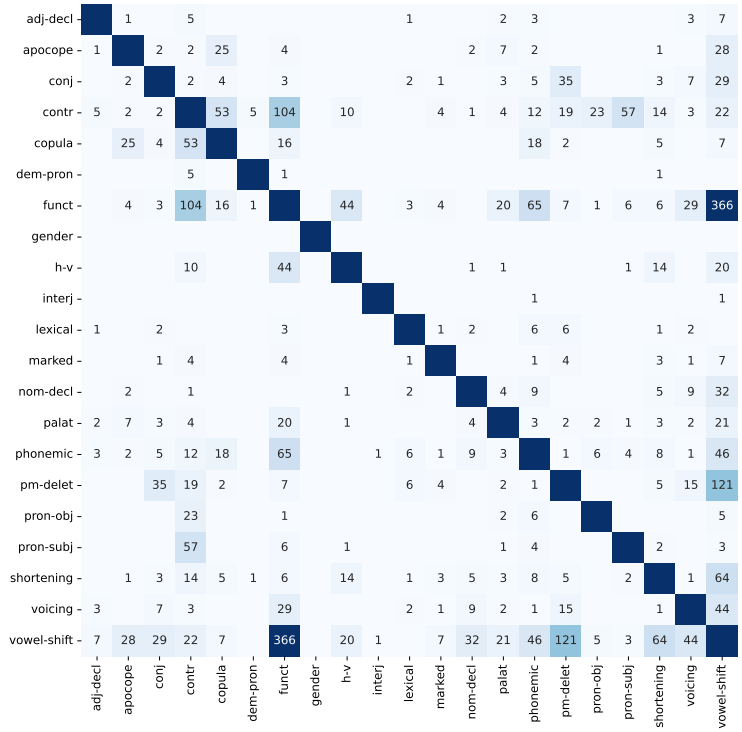


Figure 5: Co-occurrence of annotated labels.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Gabriel Doyle. 2014. Mapping dialectal variation by querying social media. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 98–106, Gothenburg, Sweden. Association for Computational Linguistics.

Jacob Eisenstein. 2013. Phonological factors in social media writing. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 11–19, Atlanta, Georgia. Association for Computational Linguistics.

Jacob Eisenstein. 2015. Systematic patterning in phonologically-motivated orthographic variation. *Journal of Sociolinguistics*, 19:161–188.

Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. 2010. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1277–1287, Cambridge, MA. Association for Computational Linguistics.

Mihaela Gamăn, Sebastian Cojocariu, and Radu Tudor Ionescu. 2021. UnibucKernel: Geolocating Swiss German jodels using ensemble learning. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 84–95, Kiyv, Ukraine. Association for Computational Linguistics.

Sidney Greenbaum and Gerald Nelson. 1996. The international corpus of English (ICE) project. *World Englishes*, 15(1):3–15.

Emiliano Raul Guevara. 2010. NoWaC: a large web-based corpus for Norwegian. In *Proceedings of the NAACL HLT 2010 Sixth Web as Corpus Workshop*, pages 1–7, NAACL-HLT, Los Angeles. Association for Computational Linguistics.

René Haas and Leon Derczynski. 2021. Discriminating between similar nordic languages. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 67–75, Kiyv, Ukraine. Association for Computational Linguistics.

Dirk Hovy and Christoph Purschke. 2018. Capturing regional variation with distributed place representations and geographic retrofitting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4383–4394, Brussels, Belgium. Association for Computational Linguistics.

Naman Jain and Riyaz Ahmad Bhat. 2014. Language identification in code-switching scenario. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 87–93, Doha, Qatar. Association for Computational Linguistics.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2021. Naive Bayes-based experiments in Romanian dialect identification. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 76–83, Kiyv, Ukraine. Association for Computational Linguistics.

Taylor Jones. 2015. Toward a Description of African American Vernacular English Dialect Regions Using "Black Twitter". *American Speech*, 90:403–440.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Klaus Krippendorff. 1980. *Content Analysis: An Introduction to Its Methodology*. Sage Publications, Beverly Hills, CA, USA.

Per E Kummervold, Javier De la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. 2021. Operationalizing a national digital library: The case for a Norwegian transformer model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 20–29, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-scale contextualised language modelling for Norwegian. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 30–40, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

William Labov. 2006. *The Social Stratification of English in New York City*, 2 edition. Cambridge University Press.

Pierre Lison, Jeremy Barnes, and Aliaksandr Hubin. 2021. skweak: Weak supervision made easy for NLP. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 337–346, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Yann Mathet, Antoine Widlöcher, and Jean-Philippe Métivier. 2015. The unified and holistic method gamma for inter-annotator agreement measure and alignment. *Computational Linguistics*, 41(3):437–479.

Preslav Nakov, Marcos Zampieri, Nikola Ljubešić, Jörg Tiedemann, Shevin Malmasi, and Ahmed Ali, editors. 2017. *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*. Association for Computational Linguistics, Valencia, Spain.

Preslav Nakov, Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Shervin Malmasi, editors. 2016. *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*. The COLING 2016 Organizing Committee, Osaka, Japan.

John Nerbonne. 2009. Data-driven dialectology. *Language and Linguistics Compass*, 3(1):175–198.

Dong Nguyen and Jacob Eisenstein. 2017. A kernel independence test for geographical language variation. *Computational Linguistics*, 43:567–592.

Younes Samih, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio. 2016. Multilingual code-switching identification via LSTM recurrent neural networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59, Austin, Texas. Association for Computational Linguistics.

Helge Sandøy. 2000. Utviklingslinjer i moderne norske dialektar. *Folkemålsstudier*, 1(39):345–384.

Mike Schuster and Kuldip Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45:2673 – 2681.

Thamar Solorio and Yang Liu. 2008. Learning to predict code-switching points. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 973–981, Honolulu, Hawaii. Association for Computational Linguistics.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: A Web-based Tool for NLP-assisted Text Annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France.

Rachael Tatman. 2016. "I'm a spawts guay": Comparing the Use of Sociophonetic Variables in Speech and Twitter. *University of Pennsylvania Working Papers in Linguistics*, 22.

Hadrien Titeux and Rachid Riad. 2021. pygamma-agreement: Gamma γ measure for inter/intra-annotator agreement in Python. *Journal of Open Source Software*, 6(62):2989.

Kjell Venås and Martin Skjekkeland. 2022. dialekter i noreg i store norske leksikon på snl.no. `https://snl.no/dialekter_i_Noreg`. Accessed: 2020-09-30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen,

Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the VarDial evaluation campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 1–15, Valencia, Spain. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Dirk Speelman, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language identification and morphosyntactic tagging: The second VarDial evaluation campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 1–17, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

NorQuAD: Norwegian Question Answering Dataset

Sardana Ivanova,[1] **Fredrik Aas Andreassen,**[2] **Matias Jentoft,**[2]
Sondre Wold,[2] and **Lilja Øvrelid**[2]
[1] University of Helsinki, Department of Computer Science
[2]University of Oslo, Language Technology Group
sardana.ivanova@helsinki.fi
{fredaan, matiasj, sondrewo, liljao}@ifi.uio.no

Abstract

In this paper, we present NorQuAD: the first Norwegian question answering dataset for machine reading comprehension. The dataset consists of 4,752 manually created question-answer pairs. We detail the data collection procedure and present statistics about the dataset. We also benchmark several multilingual and Norwegian monolingual language models on the dataset and compare them against human performance. The dataset will be made freely available.[1]

1 Introduction

Machine reading comprehension is one of the key problems in natural language understanding. The question answering (QA) task requires a machine to read and comprehend a given text passage, and then answer questions about the passage. In recent years, considerable progress has been made toward reading comprehension and question answering for English and several other languages (Rogers et al., 2022).

In this paper, we present NorQuAD: the first Norwegian question answering dataset for machine reading comprehension. The dataset consists of 4,752 question-answer pairs manually created by two university students. The pairs are constructed for the task of extractive question answering aimed at probing machine reading comprehension (as opposed to information-seeking purposes), following the methodology developed for the SQuAD-datasets (Rajpurkar et al., 2016, 2018). The creation of this dataset is an important step for Norwegian natural language processing, considering the importance and popularity of reading comprehension and question answering tasks in the NLP community.

[1]https://github.com/ltgoslo/NorQuAD

In the following we detail the dataset creation (section 3), where we describe the passage selection and question-answer generation, present relevant statistics for the dataset and provide an analysis of human performance including sources of disagreement. In order to further evaluate the dataset as a benchmark for machine reading comprehension, we perform experiments (section 4) comparing several pre-trained language models, both multilingual and monolingual models, in the task of question-answering. We also compare models against human performance for the same task. We further provide an analysis of performance across the source data domain and annotation time and present the results of manual error analysis on a data sample.

2 Related Work

Cambazoglu et al. (2021) categorise QA datasets into abstractive, extractive, and retrieval-based. In *abstractive* datasets the answer is generated in free form without necessarily relying on the text of the question or the document. In *extractive* datasets the answer needs to be a part of a given document that contains an answer to the question. In *retrieval-based* QA, the goal is to select an answer to a given question by ranking a number of short text segments (Cambazoglu et al., 2021). Since NorQuAD is constructed based on extractive QA, we will here concentrate on related work in extractive QA.

The Stanford Question Answering Dataset (SQuAD) 1.1 (Rajpurkar et al., 2016) along with SQuAD 2.0 (Rajpurkar et al., 2018) which supplements the dataset with unanswerable questions are the largest extractive QA datasets for English. SQuAD 1.1 contains 100,000+ questions and SQuAD 2.0 contains 50,000 questions.

Several SQuAD-like datasets exist for other languages. The French Question Answering Dataset (FQuAD) is a French Native Reading Compre-

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 159–168

hension dataset of questions and answers on a set of Wikipedia articles that consists of 25,000+ samples for version 1.0 and 60,000+ samples for version 1.1 (d'Hoffschmidt et al., 2020). The German GermanQuAD is a dataset consisting of 13,722 question-answer pairs created from the German counterpart of the English Wikipedia articles used in SQuAD (Möller et al., 2021). The Japanese Question Answering Dataset (JaQuAD) consists of 39,696 question-answer pairs from Japanese Wikipedia articles (So et al., 2022). The Korean Question Answering Dataset (KorQuAD) consists of 70,000+ human-generated question-answer pairs on Korean Wikipedia articles (Lim et al., 2019). The Russian SberQuAD consists of 50,000 training examples, 15,000 development, and 25,000 testing examples (Efimov et al., 2020)[2]. To the best of our knowledge there are no extractive question answering datasets available for the other Nordic languages, i.e., Danish or Swedish.

3 Dataset Creation

We collected our dataset in three stages: (i) selecting text passages, (ii) collecting question-answer pairs for those passages, and (iii) human validation of (a subset of) created question-answer pairs. In the following, we will present these stages in more detail and provide some statistics for the resulting dataset as well an analysis of disagreements during human validation.

3.1 Selection of passages

Rogers et al. (2020) reported that the absolute majority of available QA datasets target only one domain with rare exceptions. To provide some source variation in our dataset, considering our limited resources, we decided to create question-answer pairs from passages in two domains: Wikipedia articles and news articles.

We sampled 872 articles from Norwegian Bokmål Wikipedia. In order to include high-quality articles, we sampled 130 articles from the 'Recommended' section and 139 from the 'Featured' section. The remaining 603 articles were randomly sampled from the remaining Wikipedia corpus. From the sampled articles, we chose only the "Introduction" sections to be selected as passages for annotation. Following the methodology proposed for the QuAIL dataset (Rogers et al., 2020) with the goal of making the dataset more complex, we selected articles with "Introduction" sections containing at least 300 words.

For the news category, we sampled 1000 articles from the Norsk Aviskorpus (NAK)—a collection of Norwegian news texts[3] for the year 2019. As was the case with Wikipedia articles, we chose only news articles which consisted of at least 300 words.

3.2 Collection of question answer-pairs

Two students of the Master's program in Natural Language Processing at the University of Oslo, both native Norwegian speakers, created question-answer pairs from the collected passages. Each student received separate set of passages for annotation. The students received financial remuneration for their efforts and are co-authors of this paper. For annotation, we used the Haystack annotation tool [4] which was designed for QA collection. An example from the Haystack annotation environment for a Norwegian Wikipedia passage is shown in Figure 1. The annotation tool supports the creation of questions, along with span-based marking of the answer for a given passage. In total, the annotators processed 353 passages from Wikipedia and 403 passages from news, creating a total of 4,752 question-answer pairs. The remaining collected passages could be used for further question-answer pair creation.

3.2.1 Instructions for the annotators

The annotators were provided with a set of initial instructions, largely based on those for similar datasets, in particular, the English SQuAD dataset (Rajpurkar et al., 2016) and the GermanQuAD data (Möller et al., 2021). These instructions were subsequently refined following regular meetings with the annotation team. The annotation instructions will be made available along with the dataset.

3.2.2 Question generation

Annotators were instructed to read the presented passages and formulate 5-10 questions for each passage. The questions should be varied in terms of wh-question type: *hva* 'what', *hvor* 'where', *når* 'when', *hvem* 'who', *hvilke* 'which', *hvordan*

[2]The datasets are presented in alphabetical order

[3]https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-4/

[4]https://github.com/deepset-ai/haystack/

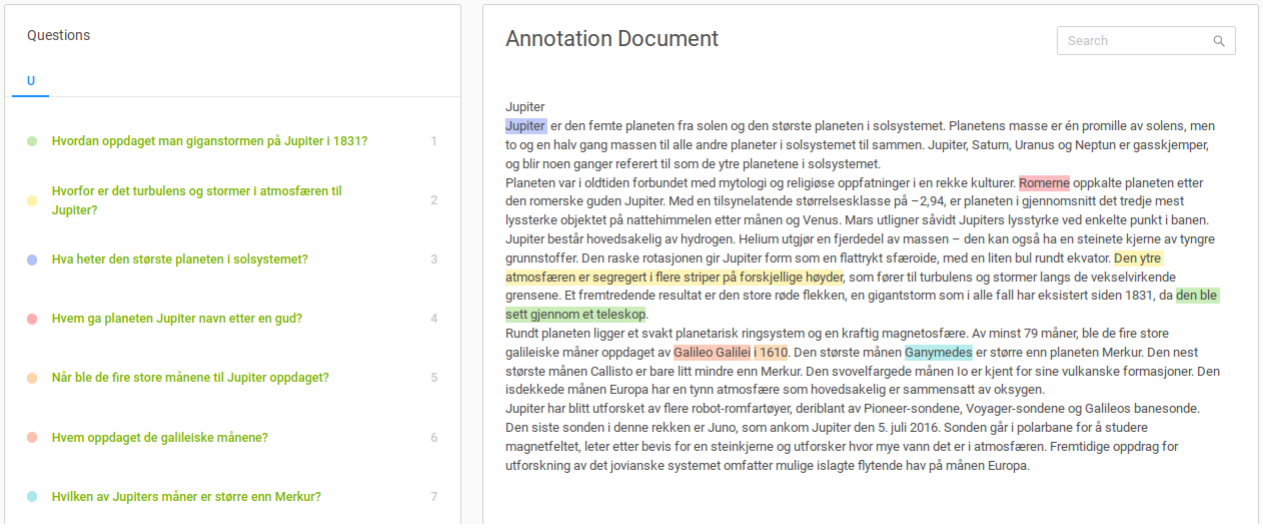


Figure 1: View of the Haystack annotation environment for a Norwegian Wikipedia document. The tool supports the creation of questions along with span-based marking of the selected answer for a document.

'how' and *hvorfor* 'why'. When formulating questions, the annotators were further instructed not to repeat or simply copy words or phrases from the passage text directly, but rather, if possible, rephrase the question. They were provided with a number of examples of types of re-phrasals, inspired by the Japanese QA dataset JaQuAD (So et al., 2022):

Syntactic variation The questions should, if possible, make use of syntactic alternations, such as the active-passive alternation:
... *John Lennon was assassinated by Mark Chapman on ..*
Q: Who assassinated John Lennon?

Lexical variation (synonymy) The questions should if possible make use of synonymy relations in re-phrasal:
... *John Lennon was assassinated by Mark Chapman on ..*
Q: Who murdered John Lennon?

Lexical variation (inference) The questions should if possible make use of inference based on lexical or world knowledge in re-phrasal:
... *John Lennon was assassinated by Mark Chapman on December 8, 1980 ...*
Q: When did John Lennon die?

Multiple sentence reasoning The questions should if possible require inference based on more than one sentence in the associated passage:
... *John Lennon was the world-famous guitarist of The Beatles. He wrote many songs, among them "All you need is love".*
Q: Who wrote "All you need is love"?

In general, the annotators were encouraged to pose difficult questions as long as they can be answered based on the information in the passage (and additional inference). The questions should in combination cover most of the passage, however, if this turned out to be difficult to balance with the requirement to pose varied questions, a priority should be given to the latter requirement. Each question should have only one answer and there are no unanswerable questions in the dataset.

3.2.3 Answer generation

The annotators were instructed to mark answers to their questions that adhere to the following main principles:

- The answer should consist of the shortest span in the original passage that answers the question.
- The answer should, however, also be a natural-sounding and a grammatically correct response to the question. As an example, for the question "When was Lennon born?" the answer text span should include the preposition "in" and not only the year "1940" if "in 1940" is indeed a span of the original text.

Question word	Wikipedia	News	Total
hva 'what'	507 (21.54%)	383 (15.97%)	890 (18.73%)
hvor 'where'	414 (17.59%)	471 (19.64%)	885 (18.62%)
når 'when'	381 (16.19%)	385 (16.06%)	766 (16.12%)
hvem 'who'	350 (14.87%)	393 (16.39%)	743 (15.64%)
hvilke 'which'	346 (14.70%)	325 (13.55%)	671 (14.12%)
hvordan 'how'	201 (8.54%)	267 (11.13%)	468 (9.85%)
hvorfor 'why'	152 (6.46%)	174 (7.26%)	326 (6.86%)
other	3 (0.13%)	0 (0%)	3 (0.06%)
Total	2354	2398	4752

Table 1: Question types distribution by question word in the dataset, broken down by data source (Wikipedia/news).

- Answers should always consist of whole words, and there should be no subword answers, such as parts of a compound or words stripped of affixes.
- Answer spans should furthermore not include span-final punctuation.
- The answers to the question should only occur once in the passage. Sometimes the same entity occurs multiple times, but it should occur only once as an answer to the relevant question.

Question word	Wikipedia	News	Total
hva 'what'	136	123	259
hvor 'where'	107	177	284
når 'when'	121	100	221
hvem 'who'	84	104	188
hvilke 'which'	95	88	183
hvordan 'how'	61	89	150
hvorfor 'why'	46	47	93
Total	650	728	1378

Table 2: Question types distribution for human validation

3.3 Dataset analysis

To understand the properties of the created question-answer pairs, we automatically categorised the whole NorQuAD dataset by question word. We provide statistics for the questions and their distribution by question word in Table 1. The "other" row in the table contains questions which we could not automatically categorise by a question word due to absence of a question word in a question or a typo in a question word. The table shows that the distribution of the various question types is fairly balanced, with the most common type being *hva* 'what' type questions (18.73% of all questions) and the least common being *hvorfor* 'why' type questions (6.86%). While the annotators were instructed to try to introduce variation in question types, the distribution of these will depend on the type of data. There are clear differences between the two data sources (Wikipedia and news), and we find that the news data contains more *hvor* 'where', *hvem* 'who' and *hvordan* 'how' type questions and less *hva* 'what' type questions than the Wikipedia portion of the dataset.

The reason for a lower occurrence of *hvorfor* 'why' and *hvordan* 'how' type questions is related to this dataset being extractive in its nature. For Norwegian, these question words require answers of a particular form, which do not occur as frequently in descriptive text as in other types of language data, such as dialogue.

It is worth noting that the overview in Table 1 does not differentiate distinct question types for questions using *hvor* as an adverb of degree, e.g. *hvor mange* 'how many', *hvor ofte* 'how often' and *hvor gammel* 'how old'. Even though *hvor* in these questions does not denote location, they are categorized as *hvor* 'where' type questions, in contrast to the GermanQuAD dataset, where *wie viele* 'how many' questions are considered a separate question type rather than just *wie* 'how' type questions (Möller et al., 2021). We found that 214 (24.18%) of our total 885 questions categorised as *hvor* 'where' type questions, are actually questions asking *hvor mange* 'how many'.

Question word	Wikipedia		News		Total	
	EM	F1	EM	F1	EM	F1
hva 'what'	66.2%	85.8%	74.0%	91.7%	69.9%	88.6%
hvor 'where'	82.2%	**94.7%**	85.9%	94.9%	84.5%	94.8%
når 'when'	81.8%	92.2%	**94.0%**	97.5%	**87.3%**	94.6%
hvem 'who'	73.8%	88.8%	83.7%	93.3%	79.3%	91.3%
hvilke 'which'	65.3%	89.1%	68.2%	88.5%	66.7%	88.8%
hvordan 'how'	72.1%	87.0%	89.9%	92.7%	82.7%	90.4%
hvorfor 'why'	**82.6%**	90.8%	91.5%	**99.0%**	87.1%	**94.9%**
Total	74.3%	89.8%	83.4%	93.7%	79.1%	91.8%

Table 3: Averaged human performance by question types

3.4 Dataset validation

In a separate stage, the annotators validated a subset of the NorQuAD dataset. In this phase each annotator replied to the questions created by the other annotator. We chose the question-answer pairs for validation at random. In total, 1378 questions from the set of question-answer pairs, were answered by validators. This provides us with a measure of human performance on a subset of the dataset. Table 2 shows the number of question-answer pairs assessed by a human, as broken down by the question types over the two data sources. It is worth mentioning that this subset is larger than the test set for which modeling results are reported in later sections. Table 3 shows the performance of the human annotators in terms of exact match and token-level F1. For the dataset as a whole we find a human performance of 79.1% exact match and a token-level F1 of 91.8%.

Exact match measures the percentage of predictions that exactly match the ground truth answers. F1 score is the harmonic mean of precision and recall. We calculate token-level F1. Both metrics ignore punctuation.

Wikipedia The annotators answered a total of 650 questions taken from the Wikipedia category. We found that human performance is 74.3% for the exact match metric and 89.8% for token-wise F1 score on average. These results are further broken down by question types in Table 3. We find that *hvilke* 'which' type questions seem to be the most difficult, with an exact match score of only 65.3%, however with considerably higher F1, indicating that for this category the precise delimitation of the answer span proves challenging. The question type with the highest F1 score of 94.7% is the *hvor* 'where', most likely due to location expressions being relatively easy to identify. The question type with the highest EM score of 82.6% is *hvorfor* 'why'.

News The annotators answered a total of 728 questions taken from the news category of the dataset. Overall exact match for this source is 83.4% with a total F1 of 93.7%. Somewhat surprisingly, the human results for this category turned out to be overall higher than for the Wikipedia category. For the news category, the *hvilken* 'which' type questions have the lowest human performance (EM 68.2%; F1 88.5%) which proved difficult also for the Wikipedia articles. The question type with the highest scores in terms of human performance for the news category are the *når* 'when' and *hvorfor* 'why' type questions.

3.5 Analysis of disagreement

As presented in Table 3, we found that human validators performed better on the news dataset than on the Wikipedia dataset. One possible reason for this is that the time of annotation affected the quality of question-answer pairs seeing that the news dataset was annotated after Wikipedia annotation. We explore this hypothesis further in Section 4.3 below. Here we examine the disagreements between the annotators during the validation phase in some more detail.

A manual inspection of the disagreements shows that there are primarily three categories of disagreements between the annotator and the validator, two of which are semantic and the last one grammatical in nature. First, there is disagreement caused by the decision of how big of a span is necessary to answer a question fully. For example, for the question *Hva er ei runebomme laget av?* 'What is a runebomme (type of drum) made

from?', the answer could be either *et dyreskinn stort sett reinskinn* 'an animal skin mainly from reindeer skin' or *et dyreskinn stort sett reinskinn spent over en oval treramme eller over en oval uthult rirkule* 'an animal skin mainly from reindeer skin, pulled over an oval wooden frame or over an oval hollowed out burl'. The first option excludes the description of how the reindeer skin is constructed, while the second one includes it.

The second category of disagreement is where an all together different span in the text is selected to answer the question. In these cases, the question is either not precisely enough formulated by the annotator to exclude other options, or the validator has been semantically imprecise in their understanding of the question. For the cases of the first type, ideally there should be no ambiguity. For example, for the question 'Where was Napoleon born?', the answer could be explicit one place in the text, as in '...He was born in Corsica..', and implicit in another '...born 15 August 1769, Corsica'. Both alternatives for the string '(in) Corsica' would be valid options, so in this case the problem is an imprecisely formulated question. For the question 'In what city is Nidarosdomen?', the answer 'Trøndelag' would be wrong, as that is a region and not a city. In that case the mistake is on the validator side, as the only correct answer would be 'Trondheim'.

The last category has to do with function words like determiners and prepositions, and whether to repeat them in the answer. Here the principles of answering with the shortest possible span but at the same time ensuring that the answers are natural sounding are in conflict. One example containing the subjunction *som* 'as' is the answer to the question *Hva arbeidet Miklos Horthy **som** fram til 1944?* 'What did MH work **as** until 1944?': *som statsoverhode i ungarn* 'as head of state in Hungary' versus the alternative answer span *statsoverhode i ungarn* which excludes the subjunction.

4 Experiments

In this section, we assess the use of the NorQuAD dataset as a benchmark for Norwegian machine reading comprehension. Given the small size of the dataset, compared to many other QA datasets, one important question to assess will be the level of performance that can be obtained with less than 5,000 question-answer pairs.

To evaluate models, we use two metrics which are used to evaluate performance on most SQuAD-like datasets: exact match and F1 score, as described in Section 3.4.

We split the NorQuAD dataset randomly into three sets: training (80%), validation (10%), and test (10%). We split datasets to the above-mentioned fractions separately for Wikipedia and news datasets to observe if performance will differ depending on domain. The test sets consist of human validated question-answer pairs, hence we may compare models' results to human performance on the same data. The human validation process is described in Section 3.4.

We establish benchmarking experiments using a set of different pre-trained language models, outlined below. Due to the small size of NorQuAD, as compared to other SQuAD-like datasets, we run all configurations five times with different random seeds and report the mean and standard deviation from these experiments. Details on selected hyperparameters are located in Appendix A.1

4.1 Baseline models

Norwegian models. We compare two monolingual transformer models based on the architecture from BERT (Devlin et al., 2019): The NorBERT2, originating from the initiative started in NorLM (Kutuzov et al., 2021), and the NB-BERT model from Kummervold et al. (2021).

Multilingual models. We further compare two multilingual models: mBERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020).

Cross-lingual augmentation. Table 1 shows that $\approx 51\%$ of the questions in our dataset use the question words *what, where* and *who*. Such questions are often answered with a named entity. Hence, we hypothesise that a lot of the annotated gold labels are named entities, and that much of the performance on SQuAD-like datasets therefore comes down to identifying the span of the correct entity in the text, which is less language specific. To investigate this, we study whether or not we can utilise another SQuAD-like dataset as a cross-lingual data augmentation step in order to increase the number of samples.

We warmup the NB-BERT model on the GermanQuAD (Möller et al., 2021) dataset for 3000 optimization steps with a batch size of 16 using a learning rate of $1e-4$. That is, we first fine-tune the NB-BERT model on the German data for a

Model	Wiki		News		All	
	EM	F1	EM	F1	EM	F1
Human*	72.65	88.84	83.61	93.43	78.13	91.14
NorBERT2	57.76 ± 1.15	71.89 ± 0.89	64.05 ± 1.27	76.93 ± 1.15	64.64 ± 1.40	77.86 ± 0.65
NB-BERT	59.74 ± 0.76	74.16 ± 1.31	67.64 ± 1.11	79.17 ± 0.92	**69.68** ± 1.21	**81.27** ± 0.73
mBERT	55.70 ± 1.67	71.21 ± 1.20	63.12 ± 2.34	73.96 ± 1.26	63.32 ± 1.58	76.00 ± 0.83
XLM-RoBERTa	54.33 ± 7.14	70.00 ± 7.76	61.72 ± 2.74	75.62 ± 2.88	64.52 ± 1.37	78.42 ± 0.97
NB-BERT$_{ger}$	**65.23** ± 1.45	**78.504** ± 1.67	**70.80** ± 1.59	**80.76** ± 1.78	68.78 ± 1.38	80.76 ± 0.62

Table 4: Results on the test set of the different domains of the NorQuAD dataset. Results are reported as means over five different random seeds with standard deviation. *Human performance is the averaged performance of the two annotators on complementary halves of the test set (10%). The model NB-BERT$_{ger}$ refers to the model with the cross-lingual augmentation from the GermanQuAD dataset. Note that the full human validated part of the dataset is larger, hence these results are not identical to those reported in Table 3.

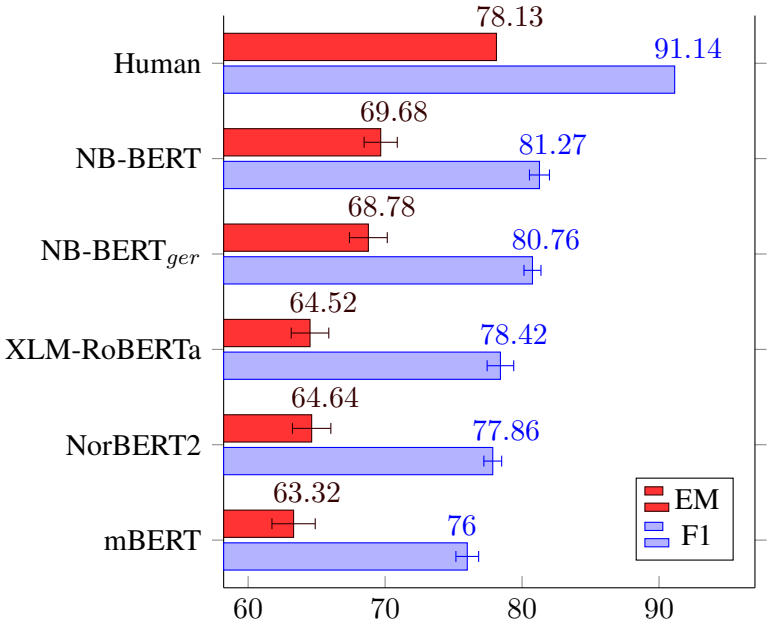


Figure 2: Performance of annotators and models on the entire test set: EM and F1 scores

limited number of steps, then switch to the Norwegian data and continue fine-tuning in the same way as for the monolingual experiments. We choose German as the augmentation language because it is typologically similar to Norwegian and since the GermanQuAD dataset has a similar annotation scheme. We chose NB-BERT as the model for cross-lingual augmentation because it was the best performing monolingual model.

4.2 Results

Table 4 shows the results of the baseline models outlined above. Overall, all models perform worse on the Wikipedia split, followed by the news split, and perform best on the total split. The best performing monolingual model, NB-BERT, performs better than both multilingual models and also has the lowest standard deviation over the different runs. We note that the performance of XLM-RoBERTa is particularly unstable. As this architecture is very similar to the models based on BERT, we did not perform a separate round of hyperparameter tuning for this model, which might explain the instability. A comparison of performance of models against human performance on all the data (both Wikipedia and news) is shown in Figure 2.

Our results indicate that it is possible to further improve the performance of a monolingual model by first warming up on the GermanQuAD dataset. This model achieves the highest score out of all the baselines on both the Wikipedia and news split with an exact match score of 65.23% and 70.80%. However, on the total split the cross-lingual data

augmentation step did not yield any added performance and performed on par with just regular fine-tuning, which points towards the cross lingual warmup being most effective for low sample scenarios. That is, when there is enough training data available, the models converge towards the same point regardless of the warmup phase. Furthermore, as the multilingual models also perform close to the monolingual ones, we interpret this as evidence towards the importance of identifying named entities for closed question answering.

Although the models perform well with respect to the relatively low sample size, as compared to other SQuAD-like datasets, there is still room for improvement when considering the human performance level.

4.3 Performance across domains and time

We noticed a difference in performance both for annotators and models on the Wikipedia and news partitions of the data sets, where the annotators and models generally performed better on the news partition. During the data collection phase, the annotators started creating question-answer pairs first for Wikipedia passages and subsequently moved on to news passages. One relevant question is therefore whether the observed difference in performance is an artifact of the way the data collection was organized. It might be that the annotators became more competent at the task over time and that the annotation for the news section is therefore more consistent and generally of a higher quality.

To evaluate whether the time of annotation affected the consistency of annotation, we measure the performance of the annotators (as obtained during the data validation stage) as well as the best performing NB-BERT model on the halves of the test dataset which were created first and the halves which were created last. These partitions measure 117/117 for Wikipedia and 119/119 for news. The results can be observed in Figure 3.

The results show that the annotators perform better on question-answer pairs created later in the annotation process both for the Wikipedia and news partitions of the dataset. We find that the average performance of the annotators on the beginning of the Wikipedia dataset is 67.29% EM and 84.99% F1, compared with 70.35% EM and 91.83% F1 on the last part of this dataset. For the beginning of the news dataset the results are

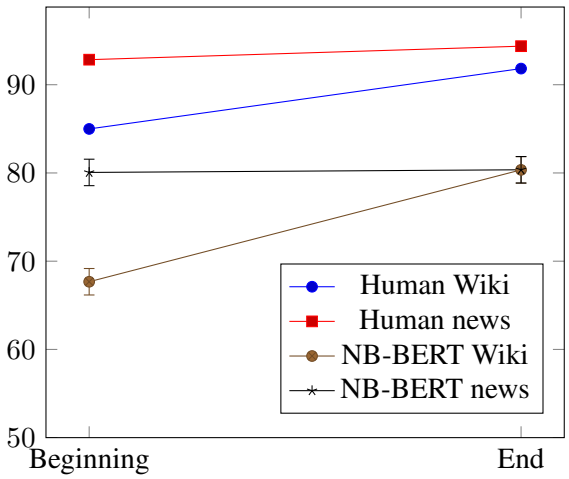


Figure 3: Performance on the beginning and end of datasets (F1 score)

81.79% EM and 92.84% F1, while the last part achieves the higher results of 85.70% EM and 94.37% F1. In total for the beginning half of the whole dataset, the performance is 74.54% EM and 88.92% F1, while the averaged performance of the annotators for the last halves are 78.03% EM and 93.10% F1.

For the models, the average performance over five runs on the beginning of the news dataset is 67.73% EM and 80.06% F1 compared to 71.35% EM and 80.36% F1 at the end. The difference in EM is within a standard deviation. On the Wikipedia dataset, however, the discrepancy is large: the average performance on the beginning is 54.98% EM and 67.67% F1 compared to 66.32% EM and 80.35% F1 at the end, with a standard deviation of ≈ 1.5 at both tails.

These results indicate a temporal effect in the data collection phase, where the annotators became more consistent in their question-answer pairs over time regardless of the domain. Since they started out with Wikipedia this effect is most noticeable in the Wikipedia category.

5 Error Analysis

In this section, we have a more detailed look at how the models performed on our dataset. We sampled 60 errors from the system output. The model in question is NB-BERT. We found that in 57% of cases there is at least some overlap between the prediction and the annotated span. In 85% of instances the predicted answers are grammatically and semantically viable phrases, but are not factually the correct answer to the current

question. Within this category there are cases of questions where the answer is the same type of entity as the one asked for in the question (e.g. a person, but the wrong person), and in other cases the entity is of a different type (e.g. a number, instead of a person).

In 13% of cases the predicted phrase could be considered a good answer to the question, but the annotator made a different selection. This means that the question must be considered ambiguous and that the system is not necessarily to blame for the error, but rather the annotation itself, because according to the annotation guidelines each question should have only one possible answer (see Section 3.2.2). Please see further discussion on this type of ambiguity in section 3.5 on analysis of disagreement between annotators.

6 Conclusion

In this paper, we presented NorQuAD—the first question answering dataset for Norwegian. We collected passages from Norwegian Wikipedia articles and a collection of Norwegian news texts and manually created over 4,700 questions. In the experiments, we fine-tuned several pre-trained language models with NorQuAD and found that the best performing model achieved 69.68% for EM and 81.27% for F1 score on the entire test set. Averaged human performance on the test set was 78.13% for EM and 91.14% for F1. The dataset and our experiments are available at `https://github.com/ltgoslo/NorQuAD`. Furthermore, we presented human validation of the part of the dataset.

We noticed that annotators and models performed better on the news dataset and we performed experiments to find out whether the time of annotation influences the performance. We measured performance of annotators and NB-BERT—the best performing model. We found that both annotators and the model perform better on the second half of the datasets meaning that the annotators got better and more consistent in their question-answer pairs over time.

While it is clear from our experiments that the number of question-answer pairs are sufficient to achieve a decent performance, improvements are certainly possible both in terms of size and data quality. To improve the dataset quality, one possible avenue for future work is to collect multiple answers for the questions as was done in SQuAD (Rajpurkar et al., 2016). Another possible extension in the future is the addition of unanswerable questions (Rajpurkar et al., 2018) to the dataset.

Acknowledgements

The project was funded by a grant from Teksthub of the University of Oslo. The project was supported in part by scholarship from Erasmus traineeship program (SMP). We are grateful for three anonymous reviewers for their many insightful comments and suggestions.

References

B. Barla Cambazoglu, Mark Sanderson, Falk Scholer, and Bruce Croft. 2021. A Review of Public Datasets in Question Answering Research. *SIGIR Forum*, 54(2).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin d'Hoffschmidt, Wacim Belblidia, Quentin Heinrich, Tom Brendlé, and Maxime Vidal. 2020. FQuAD: French Question Answering Dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1193–1208, Online. Association for Computational Linguistics.

Pavel Efimov, Andrey Chertok, Leonid Boytsov, and Pavel Braslavski. 2020. SberQuAD – Russian Reading Comprehension Dataset: Description and Analysis. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 3–15. Springer.

Per E Kummervold, Javier De la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. 2021. Operationalizing a National Digital Library: The Case for a Norwegian Transformer Model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 20–29, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-Scale Contextualised Language Modelling for Norwegian. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 30–40, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. KorQuAD1.0: Korean QA Dataset for Machine Reading Comprehension. *arXiv preprint arXiv:1909.07005*.

Timo Möller, Julian Risch, and Malte Pietsch. 2021. GermanQuAD and GermanDPR: Improving Non-English Question Answering and Passage Retrieval. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 42–50, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2022. QA Dataset Explosion: A Taxonomy of NLP Resources for Question Answering and Reading Comprehension. *ACM Comput. Surv.* Just Accepted.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting Closer to AI Complete Question Answering: A Set of Prerequisite Real Tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8722–8731.

ByungHoon So, Kyuhong Byun, Kyungwon Kang, and Seongjin Cho. 2022. JaQuAD: Japanese Question Answering Dataset for Machine Reading Comprehension. *arXiv preprint arXiv:2202.01764*.

A Appendix

A.1 Hyperparameters

The hyperparameters for the baseline models were selected based on a grid search against the validation split. The final parameters were used for all models on all splits, except for the warmup phase of the cross lingual augmentation configuration.

- $lr = 5e-5$
- $epochs = 3$
- $batch\_size\_train = 16$
- $batch\_size\_eval = 8$
- $learning\_rate\_scheduler = linear$

Extracting Sign Language Articulation from Videos with MediaPipe

Carl Börstell
Dept. of Linguistic, Literary and Aesthetic Studies (LLE)
University of Bergen (UiB)
carl.borstell@uib.no

Abstract

This paper concerns evaluating methods for extracting phonological information of Swedish Sign Language signs from video data with MediaPipe's pose estimation. The methods involve estimating i) the articulation phase, ii) hand dominance (left vs. right), iii) the number of hands articulating (one- vs. two-handed signs) and iv) the sign's place of articulation. The results show that MediaPipe's tracking of the hands' location and movement in videos can be used to estimate the articulation phase of signs. Whereas the inclusion of transport movements improves the accuracy for the estimation of hand dominance and number of hands, removing transport movements is crucial for estimating a sign's place of articulation.

1 Introduction

Sign languages – or, *signed languages* – are languages produced with gestures articulated in space and perceived visually or tactilely. Over 200 sign languages have been documented around the globe (Hammarström et al., 2022) but they are minoritized and under-researched. One challenge for quantitative research on sign languages is that they generally lack a conventionalized representation in a machine-readable form, such as phonetic transcription or orthography (see e.g., Miller, 2006; Frishberg et al., 2012; Crasborn, 2015). Following technological advances in computer vision, methods have emerged that allow a degree of form-based analysis of body movements, such as gesturing and signing, through human body pose estimation tracking of either real-time or pre-recorded video data (Pouw et al., 2020). Whereas most body pose tracking utilized in sign/gesture research used to involve either wearable devices (e.g., motion capture sensors) (Puupponen et al., 2015) or 3D cameras (e.g., Kinect) (Namboodiripad et al., 2016; Trujillo et al., 2019), thus requiring designated hardware, there are now pre-trained models that do human body pose estimation either real-time through a regular video camera or on pre-recorded video data, providing a cost-efficient alternative that has proven to be reliable in estimating human gesturing (Pouw et al., 2020). A popular tool for such analysis is OpenPose (Cao et al., 2017), which has been successfully applied in research on both sign language and gesture (Östling et al., 2018; Börstell and Lepic, 2020; Ripperda et al., 2020; Fragkiadakis et al., 2020; Fragkiadakis and van der Putten, 2021; Fragkiadakis, 2022). A tool that has become available more recently is Google's MediaPipe (Lugaresi et al., 2019), which similarly performs human body pose estimation of video data and outputs coordinates of landmarks (joints and anchor points such as eyes, nose and eyebrows).

1.1 Sign Language and Computer Vision

Previous research using OpenPose has shown that it can be used to pre-process and analyze gesture and sign language video data in terms of assessing movement (estimating articulation, holds and movement patterns) (Börstell and Lepic, 2020; Ripperda et al., 2020; Fragkiadakis et al., 2020; Fragkiadakis, 2022; Fragkiadakis and van der Putten, 2021), hand dominance (which hand is articulating more) and the number of hands involved in signing (one- vs. two-handed signs) (Östling et al., 2018; Börstell and Lepic, 2020), the place of articulation (the hands' position relative to the body) (Östling et al., 2018; Börstell and Lepic, 2020; Fragkiadakis, 2022) and even non-manual features (Kimmelman et al., 2020; Saenz, 2022). These are all basic properties of describing the form of signs and establishing the phonological structure of a sign language (Brentari, 2019). Defining the

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 169–178

start and end points of the sign articulation, excluding transport movements to and from the place of articulation, is crucial to delimit the articulation phase of a sign (Jantunen, 2015). Signs can be described as either one- or two-handed, generally evenly distributed in any sign language lexicon (Börstell et al., 2016), and two-handed signs can be further divided into *unbalanced* signs with a single active articulator (the dominant hand articulating on/by the non-dominant hand) vs. *balanced* signs, for which both hands articulate simultaneously (van der Hulst, 1996; Sandler, 2006; Crasborn, 2011). While hand dominance is generally associated with individual handedness (whether the signer is left- or right-handed), it is crucial to know which hand is dominant in one-handed and unbalance two-handed signs to establish the place of articulation, which in itself can be meaningful through iconic mappings, e.g., the head being associated with concepts relating to cognition (Börstell and Östling, 2017; Östling et al., 2018; Börstell and Lepic, 2020). The number of hands in signs has also been found to be iconically linked to plurality, such that two-handed signs are more likely to denote plural concepts (Lepic et al., 2016; Börstell et al., 2016; Östling et al., 2018).

1.2 Aims

In this paper, I evaluate methods of analyzing videos from the Swedish Sign Language online dictionary (Svenskt teckenspråkslexikon, 2023) with MediaPipe. The methods aim at extracting basic information about the articulation and sign form, which can aid quantitative research on sign languages relating to phonology and form–meaning mappings. Specifically, the aim is to evaluate methods for estimating the **articulation phase** of signs (§3.1), which can inform further analyses of sign form, and classifying signs as either left- or right-handed as **hand dominance** (§3.2) and one- or two-handed in terms of **number of hands** articulating (§3.3). Based on the hand dominance estimation and segmentation of the articulation phase, the sign's main **place of articulation** (§3.4) is estimated relative to the body.

2 Methodology

2.1 Retrieving and Processing Sign Videos

Using data from the Swedish Sign Language online dictionary (Svenskt teckenspråkslexikon, 2023) containing information about the hand dominance, number of hands and sign location for the over 20,000 signs in the database, a subset of 1,292 non-compound signs was sampled to represent a diverse set of signers in the videos (including left- and right-handed signers) and different places of articulation. Non-compounds were selected to limit each sign to a single main place of articulation and avoid combination of multiple, phonologically different elements (cf. Lepic, 2015). The sampled signs were downloaded with the `signglossR` package (Börstell, 2022) and then analyzed with the Python (3.10.5) implementation of MediaPipe (`mediapipe` 0.8.10.1), together with OpenCV (`opencv-python` 4.6.0.66) and NumPy (`numpy` 1.23.1) (Harris et al., 2020). Each video is analyzed frame by frame using the pose model estimating major landmarks on the body, represented visually in Figure 2 using the one-handed sign TAXI. The sampled sign videos vary between 35 and 312 frames in total (mean = 83, *SD* = 33) – some videos are recorded in 50 frames per second (fps), others at 25 fps. Of the 1,292 sampled videos, 43 (3.3%) show left-handed signers, the rest right-handed signers, and 567 (43.9%) involve a one-handed sign (1h), whereas 725 (56.2%) are two-handed, of which 338 are unbalanced (2h unbalanced) and 387 are balanced (2h balanced). The distribution of places of articulation is shown in Table 1.[1]

Location	***n***	**%**
head	469	36.3%
torso	184	14.2%
hand/arm	397	30.7%
neutral	155	12.0%
low	87	6.7%

Table 1: Places of articulation in sample.

2.2 Normalizing MediaPipe Outputs

A total of 107,955 frames from 1,292 videos were analyzed with MediaPipe. The output was further processed using R (4.2.2) and the packages `tidyverse` (Wickham et al., 2019), `pracma` (Borchers, 2022), `scales` (Wickham and Seidel, 2022), `slider` and (Vaughan, 2021), and graphics were created with packages `ggbeeswarm` (Clarke and Sherrill-Mix, 2017), `ggchicklet` (Rudis, 2022), `ggforce` (Ped-

[1]Locations are more fine-grained in the dictionary database, but are lumped into five major categories here.

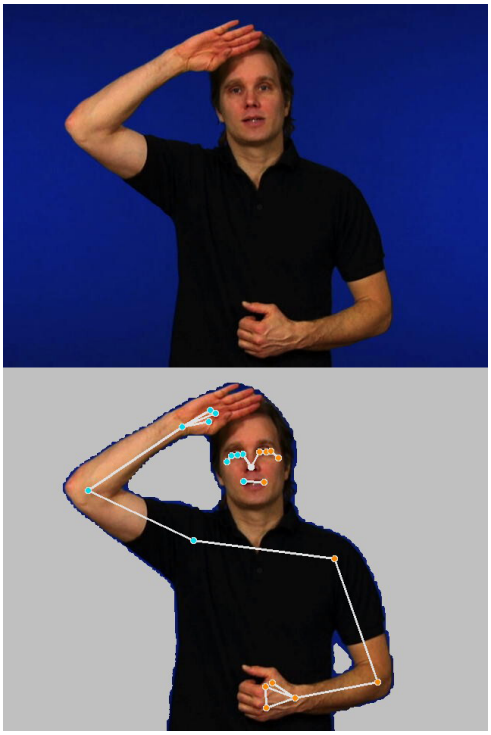


Figure 1: The sign TAXI (Svenskt teckenspråkslexikon, 2023, 1) (top) with the MediaPipe pose estimation visual output (bottom).

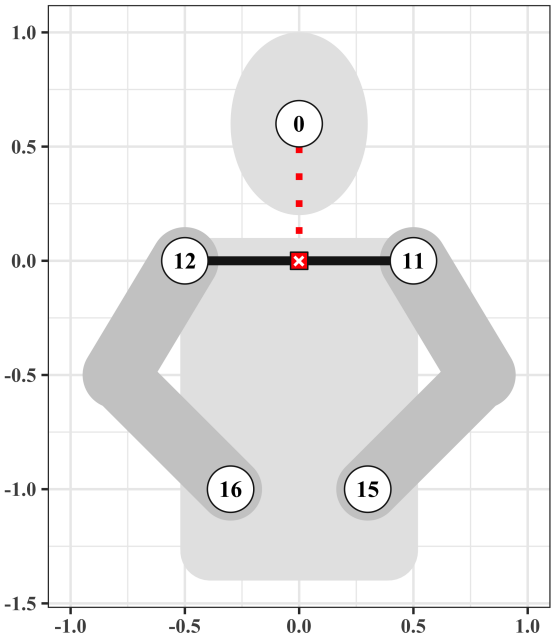


Figure 2: Relevant MediaPipe landmarks numbered, the normalized size based on the mean distance between shoulder landmarks scaled to 1, and origo set to the halfway point ("X" mark).

ersen, 2021), `ggrepel` (Slowikowski, 2022), `xtable` (Dahl et al., 2019).[2]

Only five out of the 33 landmarks of the pose estimation model were included in the further analysis, yielding a total of 539,775 datapoints, each representing a landmark estimation in a single frame. The five landmarks selected are shown in Figure 2: 0 represents the nose, 11 and 12 the left and right shoulders, and 15 and 16 the left and right wrists. The coordinate outputs from MediaPipe are scaled to 0 to 1 for both x and y. Based on the methods of Östling et al. (2018) and Fragkiadakis and van der Putten (2021), coordinates are normalized based on the mean distance between the shoulders within a sign and adjusted to an origo set at the halfway point between the mean position of the two shoulders – the red square with a white "X" in Figure 2. The coordinates were rescaled such that the distance between the shoulders equals to 1 to normalize across signers of different size, and the distance between landmark 0 and origo equals .6, to approximate the proportions of the human body.

[2]The full data set and code can be found at: `https://osf.io/x3pvq/`.

2.3 Estimating Articulation

For each sign, the articulation phase was estimated based on the movement of the two hands (or, rather, wrists) represented by landmarks 15 and 16. For each hand, the Euclidean distance traveled between each frame transition was calculated and summed into a total distance traveled. The distance traveled was smoothed into a rolling average of ±2 frames. The smoothed distance traveled data was analyzed for peaks using the `pracma::findpeaks()` function, set to look for two peaks at least 8 frames apart. These peaks represent the highest points of articulation speed, assumed to occur to and from the articulation phase – i.e., transport movements. Then, the sequence between the two peaks identified was analyzed in isolation with the same function, but with inverted values to detect valleys – assumed to represent sign holds as onset/offset in syllables (Brentari, 2019) – and set to up to 6 peaks with at least 5 frames apart. The first (inverted) peak was defined as the start frame of the articulation phase, and the last (inverted) peak was defined as the end frame. If no (inverted) peaks were identified, the

start and/or end frames were defined as the first and last original (positive) peaks, respectively. If there were less than 10 frames between the start frame and the end frame, the end frame was extended to 10 frames after the start frame. Figure 3 illustrates the original signal of the total distance traveled by the hands in the sign TAXI in grey, the smoothed signal in black, with the identified peaks as vertical, black lines, and the inverted smoothed signal between peaks as a dashed, red line, with the inverted peaks identified as vertical, red lines.

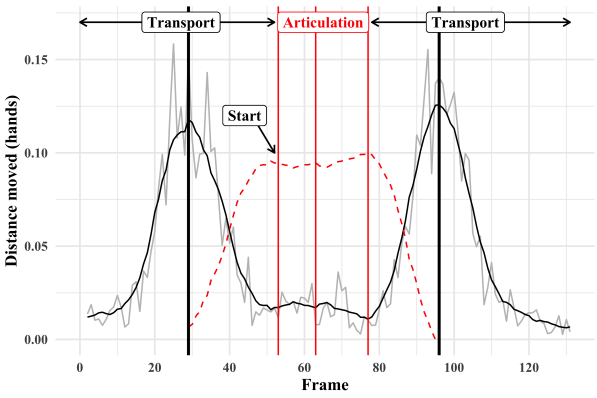


Figure 3: Distance traveled by the hands as a raw (grey) and smoothed (black) signal in the sign TAXI. Black lines show peaks in movement. The dashed, red curve is the inverted signal between peaks with lines representing peaks identified. First inverted peak is estimated start frame.

2.4 Estimating Hands

In order to estimate hand movements and locations reliably, it is important to establish which of the two hands is articulating in a sign, particularly for one-handed signs and unbalanced two-handed signs, for which the articulation is not symmetrical across the two hands. The estimation used here is simply comparing the distance traveled between the two hands: if the distance traveled by the right hand is equal to or greater than that of the left hand, the right hand is estimated to be the dominant hand, otherwise the left hand is estimated. This estimation is performed twice for each sign video: first with the distance traveled across all frames of the video (full method), then with the distance traveled within the estimated articulation phase only (short method).

Estimating the number of hands used in a sign is somewhat more complicated, as the relative difference in movement across the two hands can vary a lot, especially when a non-articulating hand can still be moving because of general body motion or readjustments (changing rest position, grooming/scratching, etc.). Östling et al. (2018) used a factor of 3 as the cut-off point between one- and two-handed signs when analyzing sign language data with OpenPose: if one hand traveled over three times the distance of the other hand, the sign was estimated to be one-handed. However, one difference between the study by Östling et al. (2018) and this one is that they calculated an extrapolated position of the hands extended from the estimated wrist position, which could lead to differences in the distance traveled. In this paper, I evaluate the accuracy of different relative factors in the distance traveled by the two hands, ranging from 1 (equal distance) to 4 (four times the distance of the other hand). This estimation is also performed twice for each sign video: first with the distance traveled across all frames of the video (full method), then with the distance traveled within the estimated articulation phase only (short method).

The estimation of place of articulation is heavily dependent on an accurate classification of hand dominance, at least for one-handed signs. In this paper, the estimation of place of articulation is made on the basis of the location of the estimated dominant hand. Since several of the locations (see Table 1) are potentially overlapping and may display internal differences – e.g., signs articulated around the head may be high or low and right or left relative to the head – the main aim here is to estimate sign height, that is the location on the *y* axis relative to origo. This estimation of place of articulation is done three times for each sign video: first using the mean coordinates of the estimated dominant hand across all frames of the video (full method), secondly, using the mean coordinates of the estimated dominant hand within the estimated articulation phase only (short method), and lastly using the coordinates of the estimated dominant hand of the estimated start frame only (start method).

3 Results

3.1 Articulation Phase

Using the peak estimation method on the distance traveled of the two hands, two main peaks were identified in all 1,292 sign videos. These peaks define the segment of the sign video that is fur-

ther analyzed for inverted peaks representing sign holds, when the hands are mostly stationary. For 47 (3.6%) out of 1,292 signs, no inverted peaks could be identified, in which case the original peaks were used as a proxy, and for 639 (49.5%) signs only a single inverted peak was found, in which case this is defined as the start frame. For 294 (22.8%) signs, the distance between start and end frames was less than 10 frames, resulting in the end frame being extended to 10 frames after the start frame. For the purpose of estimating place of articulation, the most important estimation is the initial hold phase at the beginning of the articulation phase, and with the current method of estimating this phase, 96.4% of the signs analyzed had an identified inverted peak between the transport movement peaks. Figure 4 illustrates the total distance moved by the hands across all sign videos, with vertical lines showing the mean relative locations of peaks and inverted peaks.

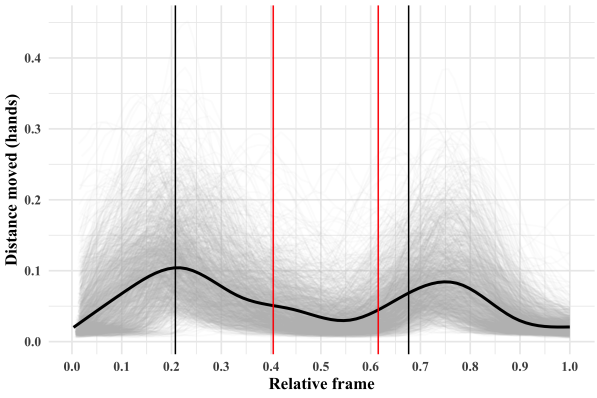


Figure 4: Distance traveled by the hands as a raw (grey) and smoothed (black) signal across all signs. Black lines show mean relative position of peaks in movement. The red lines show mean relative position of (inverted) peaks identified.

The accuracy of this method cannot be evaluated on its own without a manual annotation of each individual sign video's observed start and end points of the articulation phase. However, the method can be evaluated indirectly in the following sections, in terms of how useful the segmentation is for accurately estimating other form features of the signs, and the method will thus be discussed in more depth later.

3.2 Hand Dominance

The estimation of hand dominance was based on a simple comparison of the distance traveled by the left and right hands: if the distance traveled by the right hand is greater or equal to that of the left hand, the right hand was estimated to be the dominant hand – defaulting to the right hand for equal distances is motivated by the general right-handedness bias. The relative distance comparison was made across all frames (full method) and the frames within the estimated articulation phase only (short method).

Table 2 and Figure 5 show the accuracy of the two methods in classifying left- and right-dominant sign videos based on the actual handedness of the signers in the lexical database. The results show that the full method performs better than the short method, but both methods have a similar precision on left- and right-dominant signs.

Method	Hand	Precision	Recall	F_1
Full	left	0.81	0.88	0.85
Full	right	0.81	0.81	0.81
Short	left	0.72	0.72	0.72
Short	right	0.72	0.72	0.72

Table 2: Precision, recall and F_1 of hand dominance estimation with full and short methods.

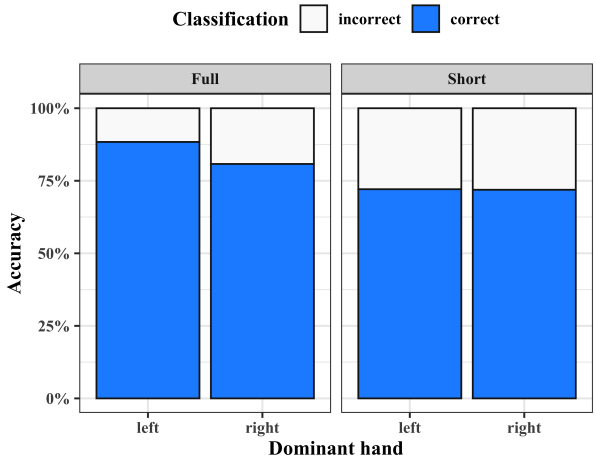


Figure 5: Accuracy of hand dominance estimation with full and short methods.

Figure 6 shows the accuracy of hand dominance estimation across different sign types with regard to the number of hands articulating: one-handed signs (1h) and two-handed signs (2h; unbalanced and balanced). The full method performs better across all three sign types, but unsurprisingly the balanced two-handed signs are approximately at chance level for both methods. The reason for this is that balanced two-handed signs are gener-

ally symmetrical in terms of both hands articulating either mirrored or alternating movements, and the hands would thus be expected to have approximately the same total distance traveled. Consequently, defining hand dominance is less important for balanced signs, since the two hands are generally symmetrical.

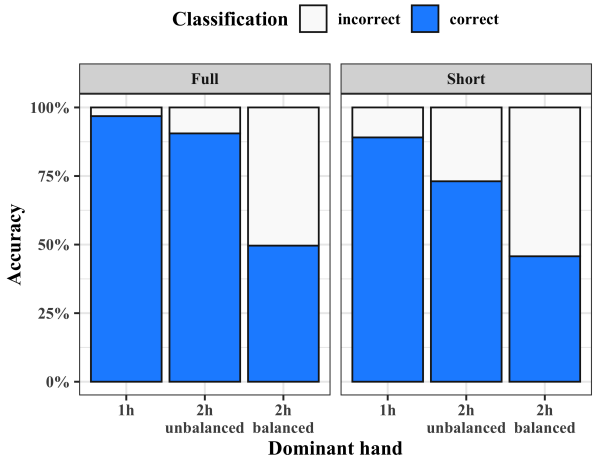


Figure 6: Accuracy of hand dominance estimation with full and short methods by sign type.

3.3 Number of Hands

The number of hands involved in each sign video was estimated by comparing the relative distance traveled between the two hands to see whether one hand traveled farther than the other hand by a factor between 1 and 4. In a previous study using OpenPose data, Östling et al. (2018) used a factor of 3 to estimate the number of hands (whether one- or two-handed). Here, the factor is increased by 0.1 increments to evaluate what the best cut-off point is for this data set. Figure 7 shows the F_1 scores for one- and two-handed signs across all factor increments for both methods, with the mean F_1 as a thicker, black line. The figure demonstrates that the best performing factor is 1.7 for the full method and 1.8 for the short method, and that the full method once again performs better overall. Table 3 shows the accuracy of classification for the best performing factors for each method.

Figure 8 shows a confusion matrix of the classification of one- and two-handed signs across the three sign types: one-handed and two-handed (unbalanced and balanced). Both methods perform relatively well with one-handed signs and balanced two-handed signs, but the unbalanced two-handed signs are particularly problematic for the

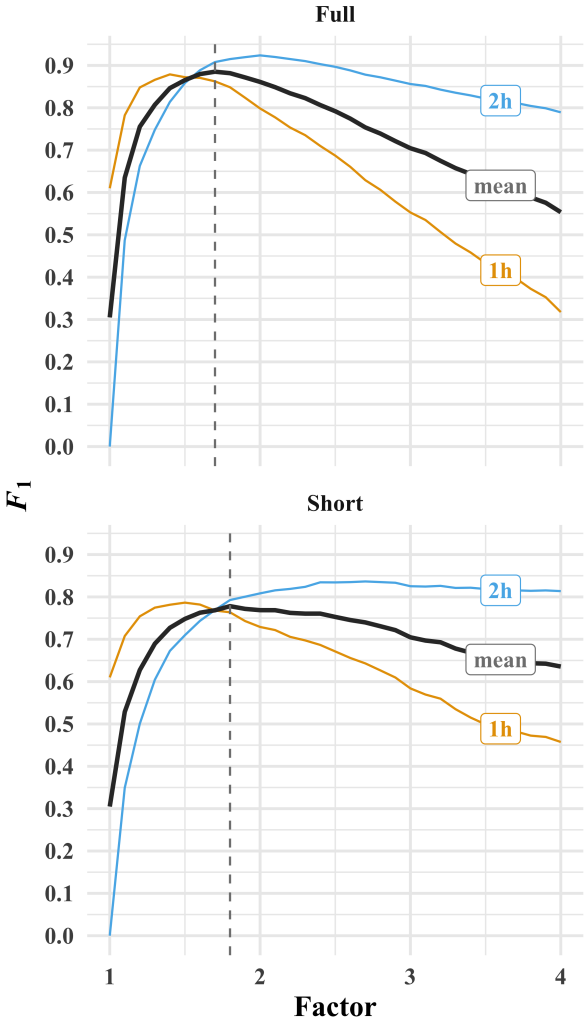


Figure 7: F_1 of number of hands estimation with full and short methods. Yellow line shows one-handed signs only, blue line shows two-handed signs only, black line shows the combined mean. Dashed, vertical black line shows the top performing factor for each method.

short method. It is unsurprising that this category poses some problems, seeing as it is an in-between sign type phonologically (cf. van der Hulst, 1996; Sandler, 2006; Crasborn, 2011), in that it has a single hand actively articulating (like one-handed signs) but two hands involved in the sign (like balanced two-handed signs).

3.4 Place of Articulation

The place of articulation of the signing for each sign video was estimated using three methods: the full method, including the mean coordinates of the estimated dominant hand across all sign frames; the short method, including the mean co-

Method	#	Fct	Precision	Recall	F_1
Full	1h	1.7	0.89	0.84	0.86
Full	2h	1.7	0.89	0.93	0.91
Short	1h	1.8	0.78	0.75	0.76
Short	2h	1.8	0.78	0.81	0.79

Table 3: Precision, recall and F_1 of number of hands estimation with full and short methods using the top performing factor for each method.

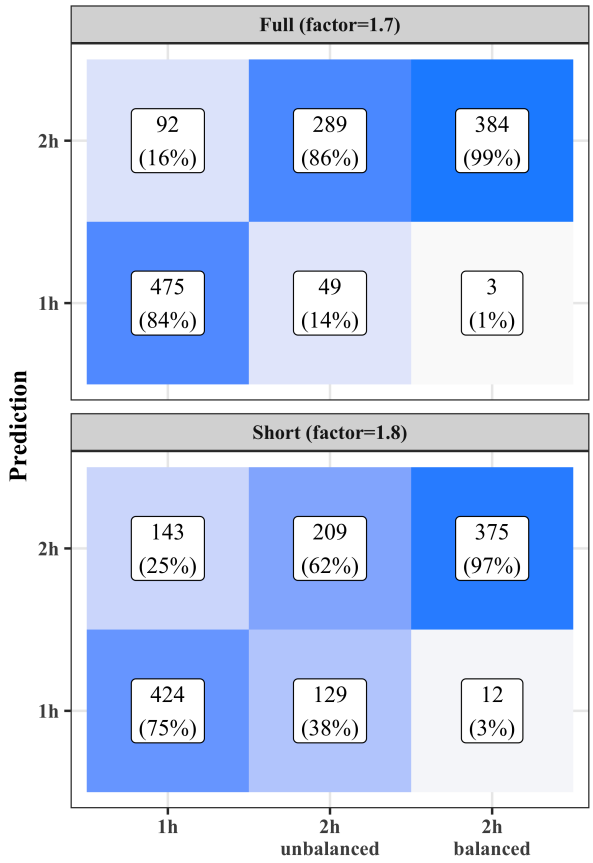


Figure 8: Confusion matrix of number of hands estimation with full and short methods, with absolute numbers and accuracy (%) for each category.

ordinates of the estimated dominant hand only for the frames inside the estimated articulation phase; and the start method, including the coordinates of the estimated dominant hand only for the estimated start frame, i.e., the first inverted peak (sign hold) inside between the transport movement peaks. Figure 9 shows the location of the estimated dominant hand relative to the signer's body across the known places of articulation for the three methods. The figure illustrates that the short and start methods perform much better than the full method. The full method conflates the hand location across the entire sign video, which means that rest positions and transport movements will always be included, and thus the estimated places of articulation are quite uniform across the actual locations as coded in the lexical database. With the short and start methods, there are visible differences in the estimated places of articulation across actual locations, which also reflect the actual locations of the signs in the lexical database – e.g., signs with a known place of articulation by the head are visibly higher up than the others. This pattern is also visible in Figure 10, which simplifies the comparison by looking at the height of the estimated place of articulation. Here, there is a much clearer – and accurate – difference across the known sign locations, showing that the short and start methods outperform the full method.

4 Conclusions

In this paper, I have shown initial explorations of methods to extract basic information about articulation and sign form from sign language video data using MediaPipe.

The first step of estimating an approximate articulation phase of the sign proved to be possible for most sign videos in the data set, which turned out to be a fruitful endeavor in order to then accurately estimate the place of articulation across signs. For the purpose of estimating hand positions corresponding to a phonological place of articulation, estimating the articulation phase is crucial, since the signal is otherwise disrupted by noise from rest positions and transport movements. Being able to automatically segment the articulation phase of signs would have other obvious applications, when extracting phonological information about the actual sign (articulation) rather than contextual noise (transport and rest).

However, when estimating hand dominance and number of signs articulating, the full method, which included data from all frames in the sign video, consistently outperformed the short method, for which the data only included frames within the estimated articulation phase. It seems as though the crude method of comparing the relative distance traveled between the two hand benefits from more data than the short articulation phase provides, and that the transport movements to and from the articulation phase are in fact quite useful for magnifying the differences in distance

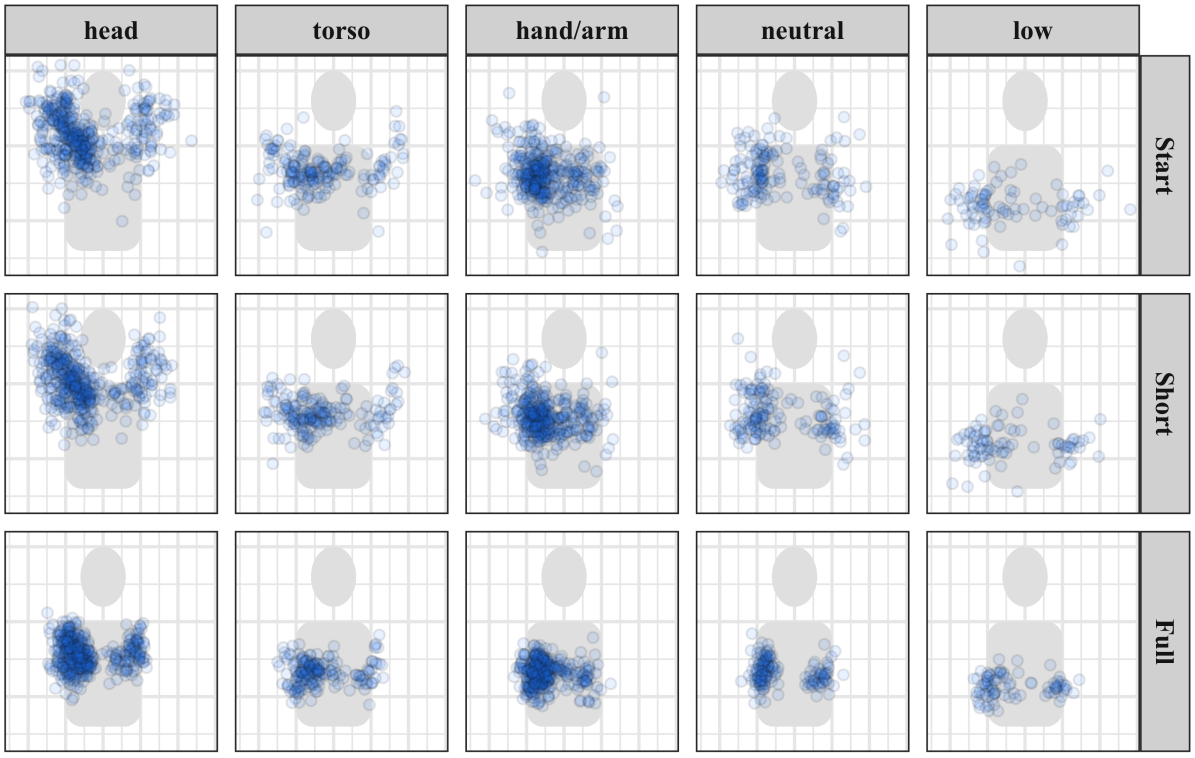


Figure 9: Estimated place of articulation across locations and three methods.

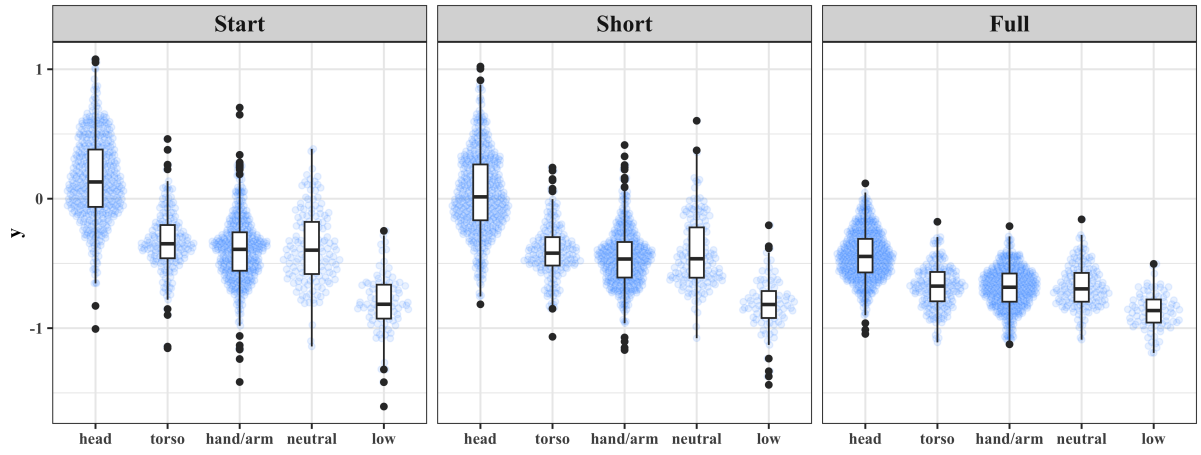


Figure 10: Estimated place of articulation as vertical sign height (y coordinates) across locations and three methods.

traveled between the two hands. This method works quite well with dictionary data here, with each video containing a single (non-compound) sign. If applied to complex/compound signs or stretches of multiple signs in succession, as in conversational data, transport movements may not be as distinct and more elaborate methods to estimate articulation phases would be necessary.

The results of this preliminary and exploratory study has demonstrated some possibilities in extracting sign language articulation from videos with MediaPipe, which can be used as a fast and cost-efficient way to analyze pre-recorded but unannotated sign language data in substantially larger quantities than would be feasible with manual annotation.

Acknowledgments

Thanks to Thomas Björkstrand for sharing Swedish Sign Language dictionary data.

References

Hans W. Borchers. 2022. *pracma: Practical Numerical Math Functions*.

Carl Börstell. 2022. Introducing the signglossR Package. In *Proceedings of the LREC2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources*, pages 16–23, Marseille, France. European Language Resources Association (ELRA).

Carl Börstell and Ryan Lepic. 2020. Spatial metaphors in antonym pairs across sign languages. *Sign Language & Linguistics*, 23(1–2):112–141.

Carl Börstell, Ryan Lepic, and Gal Belsitzman. 2016. Articulatory plurality is a property of lexical plurals in sign language. *Lingvisticæ Investigationes*, 39(2):391–407.

Carl Börstell and Robert Östling. 2017. Iconic locations in Swedish Sign Language: Mapping form to meaning with lexical databases. In *Proceedings of the 21st Nordic Conference on Computational Linguistics (NODALIDA 2017), NEALT Proceedings Series 29*, pages 221–225, Gothenburg. Linköping Electronic Press.

Diane Brentari. 2019. *Sign Language Phonology*, 1 edition. Cambridge University Press.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Erik Clarke and Scott Sherrill-Mix. 2017. *ggbeeswarm: Categorical Scatter (Violin Point) Plots*.

Onno Crasborn. 2011. The other hand in sign language phonology. In Marc van Oostendorp, Colin J. Ewen, Elizabeth Hume, and Keren Rice, editors, *The Blackwell companion to phonology, vol. 1*, pages 223–240. Malden, MA & Oxford.

Onno Crasborn. 2015. Transcription and Notation Methods. In Eleni Orfanidou, Bencie Woll, and Gary Morgan, editors, *Research Methods in Sign Language Studies*, pages 74–88. John Wiley & Sons, Ltd, Chichester.

David B. Dahl, David Scott, Charles Roosen, Arni Magnusson, and Jonathan Swinton. 2019. *xtable: Export Tables to LaTeX or HTML*.

Manolis Fragkiadakis. 2022. Assessing an Automated Tool to Quantify Variation in Movement and Location: A Case Study of American Sign Language and Ghanaian Sign Language. *Sign Language Studies*, 23(1):98–126.

Manolis Fragkiadakis, Victoria Nyst, and Peter van der Putten. 2020. Signing as Input for a Dictionary Query: Matching Signs Based on Joint Positions of the Dominant Hand. In *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, pages 69–74, Marseille, France. European Language Resources Association (ELRA).

Manolis Fragkiadakis and Peter van der Putten. 2021. Sign and Search: Sign Search Functionality for Sign Language Lexica. Publisher: arXiv Version Number: 1.

Nancy Frishberg, Nini Hoiting, and Dan I. Slobin. 2012. Transcription. In Roland Pfau, Markus Steinbach, and Bencie Woll, editors, *Sign language: An international handbook*, pages 1045–1075. De Gruyter Mouton, Berlin/Boston, MA.

Harald Hammarström, Robert Forkel, Martin Haspelmath, and Sebastian Bank. 2022. Glottolog 4.7.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Harry van der Hulst. 1996. On the other hand. *Lingua*, 98:121–143.

Tommi Jantunen. 2015. How long is the sign? *Linguistics*, 53(1):93–124.

Vadim Kimmelman, Alfarabi Imashev, Medet Mukushev, and Anara Sandygulova. 2020. Eyebrow position in grammatical and emotional expressions in Kazakh-Russian Sign Language: A quantitative study. *PLOS ONE*, 15(6):e0233731.

Ryan Lepic. 2015. The Great ASL Compound Hoax. In *Proceedings of the 11th High Desert Linguistics Society Conference*, volume 11, pages 227–250, Albuquerque, NM. University of New Mexico.

Ryan Lepic, Carl Börstell, Gal Belsitzman, and Wendy Sandler. 2016. Taking meaning in hand: Iconic motivations for two-handed signs. *Sign Language & Linguistics*, 19(1):37–81.

Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. 2019. MediaPipe: A Framework for Building Perception Pipelines. Publisher: arXiv Version Number: 1.

Christopher Miller. 2006. Sign language: Transcription, notation, and writing. In Keith Brown, editor, *Encyclopedia of Language & Linguistics*, 1988, pages 353–354. Elsevier, Oxford.

Savithry Namboodiripad, Daniel Lenzen, Ryan Lepic, and Tessa Verhoef. 2016. Measuring conventionalization in the manual modality. *Journal of Language Evolution*, 1(2):109–118.

Robert Östling, Carl Börstell, and Servane Courtaux. 2018. Visual Iconicity Across Sign Languages: Large-Scale Automated Video Analysis of Iconic Articulators and Locations. *Frontiers in Psychology*, 9:725.

Thomas Lin Pedersen. 2021. *ggforce: Accelerating 'ggplot2'*.

Wim Pouw, James P. Trujillo, and James A. Dixon. 2020. The quantification of gesture–speech synchrony: A tutorial and validation of multimodal data acquisition using device-based and video-based motion tracking. *Behavior Research Methods*, 52(2):723–740.

Anna Puupponen, Tuija Wainio, Birgitta Burger, and Tommi Jantunen. 2015. Head movements in Finnish Sign Language on the basis of Motion Capture data: A study of the form and function of nods, nodding, head thrusts, and head pulls. *Sign Language & Linguistics*, 18(1):41–89.

Jordy Ripperda, Linda Drijvers, and Judith Holler. 2020. Speeding up the detection of non-iconic and iconic gestures (SPUDNIG): A toolkit for the automatic detection of hand movements and gestures in video data. *Behavior Research Methods*, 52(4):1783–1794.

Bob Rudis. 2022. *ggchicklet: Create 'Chicklet' (Rounded Segmented Column) Charts*.

Maria Del Carmen Saenz. 2022. Mouthing Recognition with OpenPose in Sign Language. In *Proceedings of the 7th International Workshop on Sign Language Translation and Avatar Technology: The Junction of the Visual and the Textual: Challenges and Perspectives*, pages 91–94, Marseille, France. European Language Resources Association.

Wendy Sandler. 2006. Phonology, phonetics, and the nondominant hand. In Louis Goldstein, D.H. Whalen, and Catherine Best, editors, *Papers in laboratory phonology: Varieties of phonological competence*, pages 185–212. Mouton de Gruyter, Berlin.

Kamil Slowikowski. 2022. *ggrepel: Automatically Position Non-Overlapping Text Labels with 'ggplot2'*.

Svenskt teckenspråkslexikon. 2023. Svenskt teckenspråkslexikon. Place: Stockholm Published: Department of Linguistics, Stockholm University, https://teckenspraklexikon.ling.su.se/.

James P. Trujillo, Julija Vaitonytė, Irina Simanova, and Asli Özyürek. 2019. Toward the markerless and automatic analysis of kinematic features: A toolkit for gesture and movement research. *Behavior Research Methods*, 51(2):769–777.

Davis Vaughan. 2021. *slider: Sliding Window Functions*.

Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. 2019. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686.

Hadley Wickham and Dana Seidel. 2022. *scales: Scale Functions for Visualization*.

Named Entity layer in Estonian UD treebanks

Kadri Muischnek
University of Tartu
kadri.muischnek@ut.ee

Kaili Müürisep
University of Tartu
kaili.muurisep@ut.ee

Abstract

In this paper we will introduce two new language resources, two NE-annotated corpora for Estonian: Estonian Universal Dependencies Treebank (EDT, 440,000 tokens) and Estonian Universal Dependencies Web Treebank (EWT, 90,000 tokens). Together they make up the largest publicly available Estonian named entity gold annotation dataset. Eight NE categories are manually annotated in this dataset, and the fact that it is also annotated for lemma, POS, morphological features and dependency syntactic relations, makes it more valuable. We will also show that dividing the set of named entities into clear-cut categories is not always easy.

1 Introduction

Named entity recognition (NER) is an important sub-task of information extraction. In order to build a NER tagger, one first needs to annotate a corpus for named entities (NE). In this paper we introduce two NE-annotated corpora for Estonian: Estonian Universal Dependencies Treebank[1] (EDT) and Estonian Universal Dependencies Web Treebank[2] (EWT). By annotating these two resources for NE, we have aimed at broad coverage of genres, writing styles and correct vs relaxed compliance to the Estonian spelling rules.

Although there are previous NE-annotated resources for Estonian, we regard enriching existing UD corpora with NE annotation an important effort as UD annotations can support both manual annotation and help to build better NER models. We were also encouraged by the reports on similar efforts for Finnish (Luoma et al., 2020), Danish (Hvingelby et al., 2020) and Norwegian (Jørgensen et al., 2020).

In this paper, we first describe the underlying Estonian UD treebanks in Section 2.1. Section 2.2 introduces the NE categories that are distinguished in the dataset and discusses some gray areas between these classes. Corpus statistics is presented in Section 2.3 and a brief overview of related work is given in Section 3.

The NE annotations are included in the release 2.12 of Estonian UD treebanks.

2 Corpus and annotations

2.1 Estonian UD treebanks

Universal Dependencies[3] (De Marneffe et al., 2021) is an open community effort for annotating dependency treebanks using consistent annotation scheme for different human languages. Currently UD treebank collection entails nearly 200 treebanks in over 100 languages.

There are two Estonian UD treebanks: Estonian Universal Dependencies Treebank EDT and Estonian Universal Dependencies Web Treebank EWT. EDT contains ca 440,000 tokens in ca 30,000 sentences and its texts cover three central text types of normed written language: fiction, journalism and scientific writing. The text types of the treebank are not balanced: journalism with ca 270,000 tokens makes up more than half of the treebank, whereas fiction (ca 68,000 tokens) and scientific texts (ca 95,000 tokens) comprise the other half. EWT consists of texts from blog posts, online comments and discussion forums, it contains ca 90,000 tokens in ca 7000 sentences.

Universal Dependencies annotation is described thoroughly on their website[4]. For the task of NE annotation it is relevant to point out that there is a special POS-tag for proper nouns (PROPN) and a

[1] https://universaldependencies.org/treebanks/et_edt/
[2] https://universaldependencies.org/treebanks/et_ewt/
[3] https://universaldependencies.org/
[4] https://universaldependencies.org/guidelines.html

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 179–184

special syntactic relation 'flat', that is used for exocentric (headless) structures, also for multiword names. So it is relatively easy to pre-annotate the majority of NEs automatically, using these UD annotations. However, there are still some NEs that don't include a proper noun, so an annotator still has to go through the entire text carefully. Also, the exact extent of a named entity and its category have to be marked manually.

2.2 NE categories and annotation scope

Martin and Jurafsky (2021) summarize the common practice for NE annotation, noting that although a named entity is anything that can be referred to with a proper name; often also dates, times, and other kinds of temporal expressions, and even numerical expressions like prices are also included while annotating and tagging NE-s.

In our project we, at least for the time being, have annotated only "proper" NE-s, i.e. the entities that contain a proper noun or otherwise refer to a specific object like a title of a book, a film, a song etc. We have classified these entities into eight categories: persons `Per`, locations `Loc`, geo-political entities `Gep`, organizations `Org`, products `Prod`, events `Eve`, NE-s that do not fit into aforementioned categories (`Other`) and NE-s that can't be categorized due to the lack of information (`Unk`).

Often a proper noun or a title is accompanied by a headword indicating the type of the NE and thus providing valuable information. In Estonian writing, these headwords are not capitalized and they can both follow or precede the proper noun, e.g. *Tartu linn* 'Tartu city' or *romaan "Sõda ja rahu"* 'novel "War and Peace"'. Headwords of named entities are included in the annotation span, but personal titles like *härra Kask* 'mister Kask' are not.

The texts of EDT originate from the period 1998—2007 and the capitalization conventions have changed slightly during this period. *Internet* and *Sudoku* are among examples of unstable capitalization, they tend to be capitalized more in the earlier texts. Also, names of "newer" diseases like *Ebola* or *Covid* tend to be capitalized, although the language specialists suggest that a lower-case versions should be used. Of course a named entity should be annotated as such, regardless of whether it meets the spelling standard or not. On the other hand, capitalization in written Estonian is a signal that the word is a proper noun and if the text more or less follows the norms of the written language, POS tagging relies on capitalization while making the distinction between common and proper nouns. So *Internet* is a proper noun and *internet* a common noun. NE annotation, in turn, relies on POS tags, so *Internet* is a NE and *internet* is not.

Similarly, names of celestial bodies like *Maa* 'Earth' or *Kuu* 'Moon' are capitalized if referring to "a certain place in the Universe" and are treated as named entities there.

In the Estonian Web Language Treebank EWT, the texts differ from each other as to whether the author follows the norms of written language or, deliberately, does not care about them. Some writers do not use capitalization at all, others use it in an inconsistent manner. So, the POS tagging in EWT can't rely so much on capitalization but the annotator has to understand whether the reference is unique or not and NE annotations and POS tags still need to be consistent with each other.

While dividing the set of NE-s into types or categories we have put more emphasis on consistency (similar entities have to be grouped together) than on "absolute justness". So, in case that the annotators pointed out that they are persistently confused about making a clear-cut distinction between certain categories, we considered re-drawing the line. An example of `Loc` and `Gep` will be presented hereinafter.

We will now present our categories one by one.

The category `Per` includes, in addition to person names, also names of animals and imaginary creatures. Family names are annotated as `Per` even though they refer to several people, e.g. *perekond Tamm* 'Tamm family'. In internet forums, usernames are annotated as `Per`, but they are quite different from person names in general, so may be it would be a good idea to annotate them as examples of a subtype of `Per`.

The category `Loc` includes names of landscape objects like rivers or hills, and also names of man-made landscape objects like roads or settlements.

Geo-political entities `Gep` are entities that originally stand for locations, but are often represented in texts as agents – they can decide or say something etc. It is a typical case of metonymy: state or city is seen as the incarnation of its people or its governing body. This category was introduced in the annotation scheme of the Automatic Content Extraction program (ACE) (Mitchell et al.,

	news	fiction	sci	other	ewt
Per	5718	1202	1100	432	1896
Loc	2498	305	445	63	268
Gep	3324	230	442	42	318
Org	2578	47	300	73	320
Prod	1588	88	401	8	819
Event	320	5	61	1	51
Other	22	1	2	0	9
Unk	33	6	9	0	5

Table 1: Counts of named entities in treebanks

	news	fiction	sci	other	ewt
Per	2.79	1.77	1.31	4.70	2.09
Loc	1.22	0.45	0.53	0.68	0.30
Gep	1.62	0.34	0.52	0.46	0.35
Org	1.26	0.07	0.36	0.79	0.35
Prod	0.77	0.13	0.48	0.09	0.90
Event	0.16	0.01	0.07	0.01	0.06
Other	0.01	0.00	0.00	0.00	0.01
Unk	0.02	0.01	0.01	0.00	0.01

Table 2: Counts of named entities in percentile points

2003). Categorizing named entities as `Loc` or `Gep` in a consistent manner turned out to be a difficult task for the annotators, so, remaining true to our principle of prioritizing annotation consistency, we made a simplifying decision that a name of a state is always an example of `Gep`, whereas a name of a city or other settlement can be annotated as `Loc` or `Gep` depending on the context.

The decision to annotate all state names as geopolitical entities can be seen as an oversimplification, but our annotators pointed out that they kept doubting about the correct label especially in this case. Even if the word denoting a state is in a spatial case form, it is not a firm proof that it has spatial meaning and should be annotated as a place. For example, in a sentence *Raha jõudis Eestisse anonüümselt.* 'The money arrived in Estonia anonymously.' one can't infer from the text whether *Eestisse* 'in Estonia' here means the Estonian land or the Estonian state, the economic space governed by Estonian legislation.

The category `Org` is relatively straightforward. Yet there exists a grey area between organizations and products produced by those organizations. For example, the name of a newspaper can stand both for an issue of a newspaper, e.g. *in the latest Eksspress an article about elections was published* and for the editorial board of this newspaper, e.g. *Ekspress's view on elections is presented in this article*.

The category `Prod` includes man-made objects, also abstract entities such as ideas or theories. Again, the category seems to be easy at first glance, but depending on a context, a product can be presented as a location in texts: a person is in a building, a cat is in a cupboard, a fly is in a bowl. Also, products have a certain overlap with events. A movie is a product, but what about a theatre performance, taking place on a certain time and in a certain place? There is also a gray area of buildings and other man-made landscape objects, e.g. airports.

So, we have seen a few times that there exist grey areas at the borders between NE categories and, perhaps from the semantic point of view also other intersectional categories besides `Gep` would be justified. But the main objective of our work is to build a NER tagger and having too many too small NE types would hamper the NER task.

The category `Other` is used to annotate NEs that do not fit into aforementioned categories, the examples include *U3 projekt* 'the U3 project' or *Dow indeks* 'The Dow Index'. As seen from Table 1, it is the rarest of the NE categories.

The category `Unk` is used for annotating NEs which meaning is not clear. This category is more frequent in web texts, although, compared to other NE categories, it is infrequent there also. A good example of an unknown named entity originates from a fiction text describing a non-sensical lecture about *blanko-idosseeritud Pardakonossement*. Both words do not exist in Estonian, but it can be inferred from the context that *blanko-idosseeritud* is a past participle and *Pardakonossement* is a proper noun.

2.3 Annotation process

At the beginning of the project, it was clear that there was a need to annotate the entire Estonian UD treebank (approximately 530,000 tokens) in a consistent manner and also keep in mind that our created annotation should not differ drastically from the previous named entity annotation efforts.

The EDT treebank was pre-annotated automatically, based on name lists primarily including frequent person names. With the help of syntactic annotations, the extent of the named entity was at-

```
11  kus       kus       ADV    D  _                          12  advmod  12:advmod  _
12  elab      elama     VERB   V  Mood=Ind|Number=Sing       9   acl     9:acl      _
13  Eesti     Eesti     PROPN  S  Case=Gen|Number=Sing       14  nmod    14:nmod    NE=B-Gep
14  juurtega  juur      NOUN   S  Case=Com|Number=Plur       15  nmod    15:nmod    _
15  hokimehe  hoki_mees NOUN   S  Case=Gen|Number=Sing       21  nmod    21:nmod    _
16  Håkan     Håkan     PROPN  S  Case=Nom|Number=Sing       15  appos   15:appos   NE=B-Per
17  Loobi     Loop      PROPN  S  Case=Gen|Number=Sing       16  flat    16:flat    NE=I-Per
18  Kihnu     Kihnu     PROPN  S  Case=Gen|Number=Sing       19  nmod    19:nmod    NE=B-Loc
19  saarelt   saar      NOUN   S  Case=Abl|Number=Sing       20  obl     20:obl     NE=I-Loc
20  pärit     pärit     ADV    D  _                          21  advmod  21:advmod  _
21  isa       isa       NOUN   S  Case=Nom|Number=Sing       12  nsubj   12:nsubj   _
22  Paul      Paul      PROPN  S  Case=Nom|Number=Sing       21  appos   21:appos   NE=B-Per
```

Figure 1: Corpus example: annotated clause *where hockey player with Estonian roots Hakan Loob's father Paul from Kihnu island lives*

tempted to be identified, and for remaining proper names, annotations `B-Unk` (first member of the named entity) and `I-Unk` (subsequent members of the named entity) were added.

Initially, there were 3 student annotators who annotated the texts; at the first stage texts were annotated by two students, the annotations compared and the discrepancies solved. The students had different skills and availability, so eventually, one student continued to work alone. If the annotator felt that the solution was not unambiguous, he wrote a question into the log-file, which was later discussed with supervisors. Lists of annotated named entities were also compiled and reviewed together. The EWT corpus, which is smaller in size but more complex in content, was annotated by a student and then checked and corrected by supervisors. This method for annotation does not allow for calculations to assess the inter-annotator agreement measures but we believe that a multi-person, multiple-check annotated corpus is the best that could be created given limited resources.

2.4 Corpus statistics

Tables 1 and 2 show the raw and normalized NE frequencies in EDT and EWT and the distribution of NEs in different text types. EDT contains the main text classes of normed written language: newspaper texts, fiction and scientific texts. Only one text, containing example sentences from a scientific work about Estonian valency patterns, plus sentences from different news texts, belongs to the text class "other". EWT contains the text classes of user-generated content: blog posts, comments and forum texts.

In EDT, the frequencies are distributed as could be expected: newspaper texts have the highest density of NEs, fiction texts contain lot of person names. Scientific texts include references, that, somewhat unnaturally, increase the frequency of person names in them. The text class "other" does not represent normal text: the example sentences of the valency frames include person names (never pronouns or common nouns referring to a human) wherever a word denoting a human was possible, e.g. *Mary saw John*.

EWT forum texts include usernames, that are annotated as `Per` and the users also address each other using their usernames. In web forums people also discuss and rate various products, which raises the frequency of `Prod` category.

3 Related work

3.1 Estonian NE-annotated corpora

There are two previous NE-annotated corpora for present-day Estonian and one for historical Estonian. Tkachenko and colleagues (Tkachenko et al., 2013) have annotated four NE categories (persons, locations, organizations and other) in a 185,000-token dataset.

New Estonian NER dataset[5] contains ca 140,000 tokens and the annotated NEs are divided into 7 categories: persons, organizations, locations, geo-political entities, titles, products and events. In addition to "proper NEs", also dates, times, percents and currencies are annotated. During this project also Tkatchenko's dataset was re-annotated. The resulting datasets use hierarchical annotation, which we regard useful, but for the time being have refrained from using it in order to make the task easier for the annotators.

In a corpus of historical Estonian, a collection of parish court records from the 19th century (Orasmaa et al., 2022) seven NE categories are annotated: person, location, organization, location-

[5] https://github.com/TartuNLP/EstNER_new

organization, artefact, other and unknown. The parish court records make up a text type of its own, but their NE typology is similar to that of our corpora; the category 'location-organization' is essentially the same as our `Gep` and the category 'artefact' is similar to our `Prod`.

3.2 Other NE-annotated resources based on UD annotations

In the Finnish corpus (Luoma et al., 2020), six NE categories have been annotated: person, organization, location, product and event names as well as dates. In order to to avoid ambiguity-creating categories, geopolitical entities are annotated as locations, but the authors admit that for applications where the resolution of the ambiguity is not critical, there may be merit to the adoption of possibly ambiguity-creating type like geo-political entity.

In the Danish NE-annotated corpus (Hvingelby et al., 2020) four NE classes are annotated: location, organisation, person and miscellaneous, following the guidelines of the CoNLL-2003 NE annotation scheme (Sang and Meulder, 2003). They also report that it was difficult for the annotators to distinguish between locations and organizations in certain cases.

In the Norwegian UD treebank (Jørgensen et al., 2020) the categories of person, organization, location, geo-political entity, product and event have been annotated. Geo-political entities are subcategorized as either GPE with a locative sense or GPE with an organization sense. However, while annotating the corpus with those categories, annotators had some difficulties with making the distinction between the subcategories of the GPE entity types. Building on the experience of NOrNE annotation effort, we did not attempt at dividing the category of `Gep` into subcategories.

4 Conclusions and future directions

We have presented two manually annotated NER datasets for Estonian. The annotated texts represent the core text types of normed written language as well as several text types of the user-generated content of the web. The annotated NEs fall into eight categories: persons, locations, geo-political entities, organizations, products, events, other NEs that can't be classified into aforementioned categories and NEs of unknown category. The category of geo-political entities is a hybrid category between location and organization. Although we noticed that there exist also other cases of systematic metonymy besides using a location name to note the people connected with this location, we did not introduce more NE types as we did not want to divide the NEs into too many too small categories.

Obviously the next step would be building NER models using this dataset. Also, as the web treebank EWT is being developed further, annotating new genres of web texts with UD annotations, we plan to add the new texts into our dataset.

Acknowledgments

This work has been supported by the Estonian National Programme "Estonian Language Technology" via grants EKTB7 "Estonian Universal Syntax: Resources and Applications" and EKTB75 "Basic resources for semantic analysis".

We would like to thank Mihkel Rünkla and other annotators.

References

Marie-Catherine De Marneffe, Christopher D Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational linguistics*, 47(2):255–308.

Rasmus Hvingelby, Amalie Brogaard Pauli, Maria Barrett, Christina Rosted, Lasse Malm Lidegaard, and Anders Søgaard. 2020. DaNE: A named entity resource for Danish. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4597–4604, Marseille, France. European Language Resources Association.

Fredrik Jørgensen, Tobias Aasmoe, Anne-Stine Ruud Husevåg, Lilja Øvrelid, and Erik Velldal. 2020. NorNE: Annotating named entities for Norwegian. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4547–4556, Marseille, France. European Language Resources Association.

Jouni Luoma, Miika Oinonen, Maria Pyykönen, Veronika Laippala, and Sampo Pyysalo. 2020. A broad-coverage corpus for Finnish named entity recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4615–4624, Marseille, France. European Language Resources Association.

James H. Martin and Dan Jurafsky. 2021. Speech and Language Processing (3rd ed. draft). *URL: https://web.stanford.edu/ jurafsky/slp3/.*

Alexis Mitchell, Stephanie Strassel, Mark Przybocki, JK Davis, George R. Doddington, Ralph Grishman, Adam Meyers, Ada Brunstein, Lisa Ferro, and Beth Sundheim. 2003. Ace-2 version 1.0. *LDC2003T11.*

Siim Orasmaa, Kadri Muischnek, Kristjan Poska, and Anna Edela. 2022. Named entity recognition in Estonian 19th century parish court records. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5304–5313, Marseille, France. European Language Resources Association.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147. ACL.

Alexander Tkachenko, Timo Petmanson, and Sven Laur. 2013. Named entity recognition in Estonian. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 78–83, Sofia, Bulgaria. Association for Computational Linguistics.

ScandEval: A Benchmark for Scandinavian Natural Language Processing

Dan Saattrup Nielsen
The Alexandra Institute
Copenhagen, Denmark
dan.nielsen@alexandra.dk

Abstract

This paper introduces a Scandinavian benchmarking platform, `ScandEval`, which can benchmark any pretrained model on four different tasks in the Scandinavian languages. The datasets used in two of the tasks, linguistic acceptability and question answering, are new. We develop and release a Python package and command-line interface, `scandeval`, which can benchmark any model that has been uploaded to the Hugging Face Hub, with reproducible results. Using this package, we benchmark more than 100 Scandinavian or multilingual models and present the results of these in an interactive online leaderboard[1], as well as provide an analysis of the results. The analysis shows that there is substantial cross-lingual transfer among the Mainland Scandinavian languages (Danish, Swedish and Norwegian), with limited cross-lingual transfer between the group of Mainland Scandinavian languages and the group of Insular Scandinavian languages (Icelandic and Faroese). The benchmarking results also show that the investment in language technology in Norway, Sweden and Denmark has led to language models that outperform massively multilingual models such as XLM-RoBERTa and mDeBERTaV3. We release the source code for both the package[2] and leaderboard[3].

1 Introduction

In recent years, there has been a significant increase in the number of monolingual language models in the Scandinavian languages (Møllerhøj, 2020; Højmark-Bertelsen, 2021; Sarnikowski, 2021; Enevoldsen et al., 2021; Abdaoui et al., 2020; Kummervold et al., 2021; Malmsten et al., 2020; Snæbjarnarson et al., 2023), to the extent that it becomes difficult both for the practioner to choose the best model for the task at hand, as well as for language researchers to ensure that their research efforts are indeed improving upon past work.

Aside from the increasing number of models, Sahlgren et al. (2021) also emphasises that a joint Scandinavian language model is probably a better strategy for the Scandinavian countries, considering the similarity of their languages and culture. Indeed, Faarlund (2019) even argues that the Danish, Norwegian and Swedish languages are so similar that they should be considered a single language.

The languages included in the term "Scandinavian" is debatable (oxf, 2021). Following the distinction between *Mainland Scandinavian* (Danish, Swedish and Norwegian) and *Insular Scandinavian* (Icelandic and Faroese) (Haugen, 1976; Faarlund, 2019), a distinction based on mutual intelligibility and syntactical structure, we focus in this work on the Mainland Scandinavian languages, while still allowing support for the Insular Scandinavian languages. Aside from being a standard distinction, our choice is also based on experiments on the cross-lingual transfer between these two groups, which we present in Section 3.3. We will here use the term "Scandinavian" to mean the collection of all five languages, and use the Mainland/Insular distinction when applicable.

To help facilitate progress in both improving upon the monolingual Scandinavian models as well as the multilingual, we present `ScandEval`, a benchmark of Scandinavian models, along with a Python package and Command-Line Interface (CLI), and an associated online leaderboard. This leaderboard contains the results of language models benchmarked on datasets within the Mainland

[1] https://scandeval.github.io
[2] https://github.com/saattrupdan/ScandEval
[3] https://github.com/ScandEval/scandeval.github.io

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 185–201

Scandinavian languages, as described in Section 4.

Recent studies (Khanuja et al., 2021; Pires et al., 2019; Lauscher et al., 2020) have shown that multilingual models can outperform monolingual models when the languages are sufficiently similar, and also that they are worse than the monolingual models when the languages are too dissimilar. This shows that the Scandinavian languages could have something to gain by creating "local multilingual" models, rather than using the massively multilingual models such as `XLM-RoBERTa` (Conneau et al., 2020). Based on this, we test the following hypotheses:

- **Hypothesis 1:** There is a substantial cross-lingual transfer within the Mainland Scandinavian languages.
- **Hypothesis 2:** There is no notable cross-lingual transfer between the group of Mainland Scandinavian languages and the group of Insular Scandinavian languages.

To the best of our knowledge, this is the first benchmarking tool for any of the Scandinavian languages, as well as the first online leaderboard containing scores from such a tool. Our contributions are the following:

1. We construct a new question answering dataset for the Mainland Scandinavian languages, dubbed `ScandiQA`.
2. We construct a new linguistic acceptability dataset for all the Scandinavian languages, dubbed `ScaLA`.
3. We develop a Python package and CLI, `scandeval`, which allows reproducible benchmarking of language models on Scandinavian language datasets.
4. We uniformise all the datasets used in the benchmark, to enable consistent evaluation across languages and datasets. These uniformised datasets are also available on the Hugging Face Hub[4].
5. We benchmark all the Scandinavian and a selection of the multilingual language models on the Hugging Face Hub[5] on the Mainland Scandinavian datasets in the benchmark, and present all the scores in an online leaderboard.

[4] `https://huggingface.co/ScandEval`
[5] `https://hf.co`

2 Related Work

There has been a number of (non-English) NLU benchmarks published in recent years (Wang et al., 2018; Sarlin et al., 2020; Rybak et al., 2020; Ham et al., 2020a; Shavrina et al., 2020; Wilie et al., 2020; Xiang et al., 2021; Koto et al., 2020; Safaya et al., 2022; Augustyniak et al.; Khashabi et al., 2020; Ham et al., 2020b; Xu et al., 2020; Dumitrescu et al., 2021), with whom we share the same goal of advancing the state of NLP in our respective languages. Within the Scandinavian languages specifically, the SuperLim benchmark (Adesam et al., 2020) is a Swedish NLU Benchmark featuring several difficult tasks. Most of the datasets in the SuperLim benchmark only contain a test set, however.

The `XGLUE` (Liang et al., 2020) dataset is another multilingual NLU benchmark. That dataset is different from `ScandEval` in that all the training data in `XGLUE` is in English, and that the majority of the test sets are not available in any of the Scandinavian languages.

Isbister and Sahlgren (2020) present a Swedish similarity benchmark, achieved through machine translating the `STS-B` dataset from the GLUE benchmark (Wang et al., 2018). Aside from only dealing with a single task and a single language, the quality of the dataset is worse than a gold-standard corpus as a result of the translation, as the authors also point out.

3 Methodology

This section describes our benchmarking methodology in detail, including both the setup of the datasets, the evaluation procedure and the scoring of the models. We also describe how we conduct the cross-lingual transfer experiments.

3.1 Finetuning Setup

When finetuning, we enforce a learning rate of $2 \cdot 10^{-5}$ with 100 warmup steps, and a batch size of 32. If there is not enough GPU memory to finetune the model with this batch size, we halve it and double the amount of gradient accumulation, resulting in the same effective batch size. This is repeated until the batches can fit in memory.

We impose a linear learning rate schedule with intercept after $10,000$ training steps (with a training step consisting of 32 samples), and we adopt early stopping (Plaut et al., 1986) to stop the training procedure if the validation loss has not de-

creased for 90 training steps. We use the `AdamW` optimiser (Loshchilov and Hutter, 2018) with first momentum $\beta_1 = 0.9$ and second momentum $\beta_2 = 0.999$, and we optimise the cross-entropy loss throughout all tasks. Further, random seeds are fixed throughout, to ensure reproducibility.

The finetuning itself uses the `transformers` package (Wolf et al., 2020). For the named entity recognition task we use the `AutoModelForTokenClassification` class, which linearly projects the embedding from the language model encoder for each token into the entity logits for that token. For the classification tasks we use the `AutoModelForSequenceClassification` class, which linearly projects the embedding from the language model encoder to each document into the class logits for that document. Lastly, for the question answering task we use the `AutoModelForQuestionAnswering` class, which linearly projects the embedding from the language model encoder for each token, into the logits of the start and end positions of the answer for that token.

3.2 Bootstrapping Evaluation

For each model and dataset, we repeat the following procedure 10 times, which generates a score for each model and dataset combination: (a) Fix a random seed unique to the given iteration; (b) Finetune the model on the training set; (c) Evaluate the model on a bootstrapped (i.e., sampling with replacement) version of the test set. The evaluation score is then the mean μ of these scores, along with a 95% confidence interval I_{10}, computed as

$$I_N := \mu \pm \frac{1.96}{N-1} \sum_{i=1}^{N} \texttt{score}_i. \qquad (1)$$

The combination of varying the random seeds as well as using bootstrapped test datasets ensures that we capture the noise coming from both the random initialisation of the added layers to the model as well as the noise in the test set, resulting in a more reliable confidence interval of the true mean for each model and dataset combination.

To aggregate these scores across all datasets, we firstly compute the *language-specific task scores* for each (model, language, task) triple, which is the mean of the scores of the model on the tasks of the language.[6] From these language-specific scores we next compute the *language score* for each (model, language) pair as the mean of the language-specific task scores across all the tasks. A final `ScandEval` score is computed as the average of the language scores, to emphasise the training of Scandinavian models rather than monolingual ones.

3.3 Cross-lingual Transfer

To test Hypothesis 1 and 2, stated in Section 1, we introduce a way to measure the "joint cross-lingual transfer" of a group of languages, by which we mean an aggregate of the cross-lingual transfer between any two languages in the group.

To do this, we first introduce a control group of non-Scandinavian languages: English, German, Dutch, Finnish, Russian and Arabic. By considering the combined set of languages in the control group and the Scandinavian languages, we aim to find the best split of these languages into two groups: a ScandEval benchmark group and a non-benchmark group. The "goodness" of a split is measured by benchmarking a "representative" model from each language on datasets in each of the benchmark languages and measuring the quality of the two-cluster clustering of these benchmarking values.

As an example, if Danish and Swedish constitute the benchmark group and the rest of the languages are in the non-benchmark group, we would benchmark the representative models from each language on the Danish and Swedish part of ScandEval, and then compute the F-statistic of the clustering $\{\{\texttt{da}, \texttt{sv}\}, \{\texttt{no}, \texttt{is}, \texttt{fo}, \ldots\}\}$ with these benchmarking values, computed as the ratio of the between-group variance to the within-group variance.[7] We can then compare this F-statistic to the F-statistic of the clustering where the benchmark group consists of Danish and Norwegian, for instance.

As for picking a representative model for each language, we found pretrained language models of roughly the same size on the Hugging Face Hub, each of which has been pretrained on solely monolingual data. We note that no Faroese language model exists, so for that language we do not include any model but still include Faroese benchmarking

[6] This mean is only non-trivial for the Norwegian language for the named entity recognition task and the linguistic acceptability task, as these tasks are available in both Norwegian Bokmål and Norwegian Nynorsk.

[7] Technically speaking, we get an F-statistic for each language in the benchmarking group, but we just use the mean of these F-statistics.

datasets when Faroese is part of the benchmarking group. See the full list of models in the appendix.

We can then restate our first hypothesis as the mainland Scandinavian languages are all in the best-performing benchmark group, and our second hypothesis as the Insular Scandinavian languages are not in the best-performing benchmark group.

3.4 Uniform Benchmarking Datasets

As we are interested in comparing the performance of the models across languages, we ensure that all the datasets used in the benchmark are of the same format and the same size.

We aimed to choose a training data size that would be a balance between being able to differentiate between the models and being able to benchmark the models in a reasonable amount of time. We benchmarked the same models as in Section 3.3 on truncations of named entity recognition datasets, sentiment classification datasets and linguistic acceptability datasets. Based on these results we qualitatively found that using 1,024 training samples allowed for both differentiation between the models and being able to benchmark the models in a reasonable amount of time. Figures 1 and 2 show the trade-off between differentiation and benchmarking speed, covering the `AngryTweets` dataset (Pauli et al., 2021). The remaining plots for the other datasets can be found in the appendix.

Another benefit of using a small training dataset is that it emphasises the importance of the pretrained weights of the models, rather than the finetuning process. Further, we wanted the test dataset to be as large as possible, to ensure more robust evaluations of the models, which led to the choice of 2,048 test samples based on the number of available samples in the smallest dataset. Lastly, the validation set was chosen to be 256 samples, to allow for a reasonable evaluation during training, while not being too time-consuming. All of these datasets with their splits are available on the Hugging Face Hub.

4 ScandEval Tasks

To properly evaluate the performance of a pretrained model, we ideally need to evaluate it on many diverse tasks. Unfortunately, the Scandinavian languages do not have many openly available datasets for many downstream tasks.

To address this, we construct two new Scandinavian datasets, `ScaLA` and `ScandiQA`, being *Linguistic Acceptability* (`LA`) and *Question Answering* (`QA`) datasets, respectively. These new tasks are supplemented by existing benchmarking datasets within *Named Entity Recognition* (`NER`) and *Sentiment Classification* (`SENT`). Aside from downstream performance of these tasks, we also benchmark the inference speed of each model. We describe all of these in more detail in the subsections below.

4.1 Named Entity Recognition

For the NER task we use the four classes used in CONLL (Tjong Kim Sang and De Meulder, 2003): `PER`, `LOC`, `ORG` and `MISC`, corresponding to person names, locations, organisations and miscellaneous entities.

Since this is a token classification task and that the language models usually use different tokenisers, we have to ensure a uniform treatment of these as well. We tokenise the documents using the pretrained tokeniser associated to the model that we are benchmarking, and to ensure consistency of the evaluation we replace all but the first token in each word with the empty entity `O`. For instance, if the word "København" with the `LOC` tag is tokenised as ["Køben", "havn"], then we would assign the labels `LOC` and `O` to these tokens. This ensures that we maintain the same number of (non-empty) labels per document.

In terms of evaluation metrics, we use the micro-average F1-score, which is standard for NER. We also report a *no-misc score*, which is the micro-average F1-score after we replace the `MISC` class in the predictions and labels with the "empty label" `O`. This *no-misc score* is not used in any of the aggregated scores and is purely used for comparison purposes on the individual datasets.

For Danish we use the `DaNE` dataset (Hvingelby et al., 2020), being a NER tagged version of the Danish Dependency Treebank (Kromann and Lynge, 2004). `DaNE` is already in the CONLL format, so we perform no preprocessing on the data.

For Norwegian we use the Bokmål and Nynorsk `NorNE` datasets (Jørgensen et al., 2020), also being NER tagged versions of the Norwegian Dependency Treebanks (Øvrelid and Hohle, 2016). Aside from the `PER`, `LOC`, `ORG` and `MISC` tags, these also include `GPE_LOC`, `GPE_ORG`, `PROD`, `DRV` and `EVT` tags. We convert these to `LOC`, `ORG`, `MISC`, `MISC` and `MISC`, respectively.

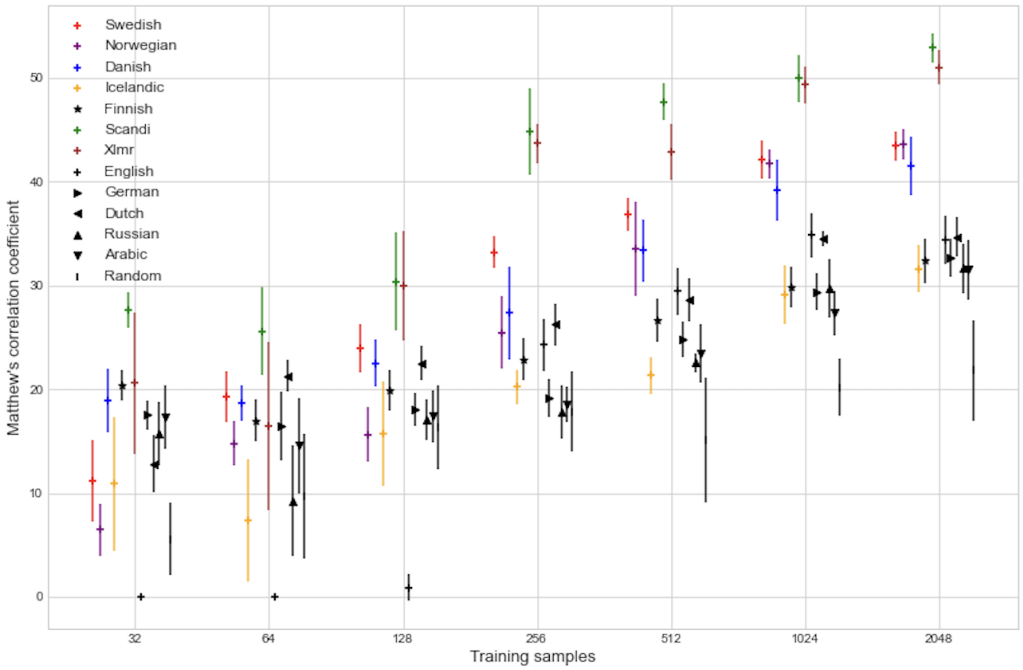


Figure 1: Plot showing the performance of different models on the `AngryTweets` dataset with varying number of training samples.

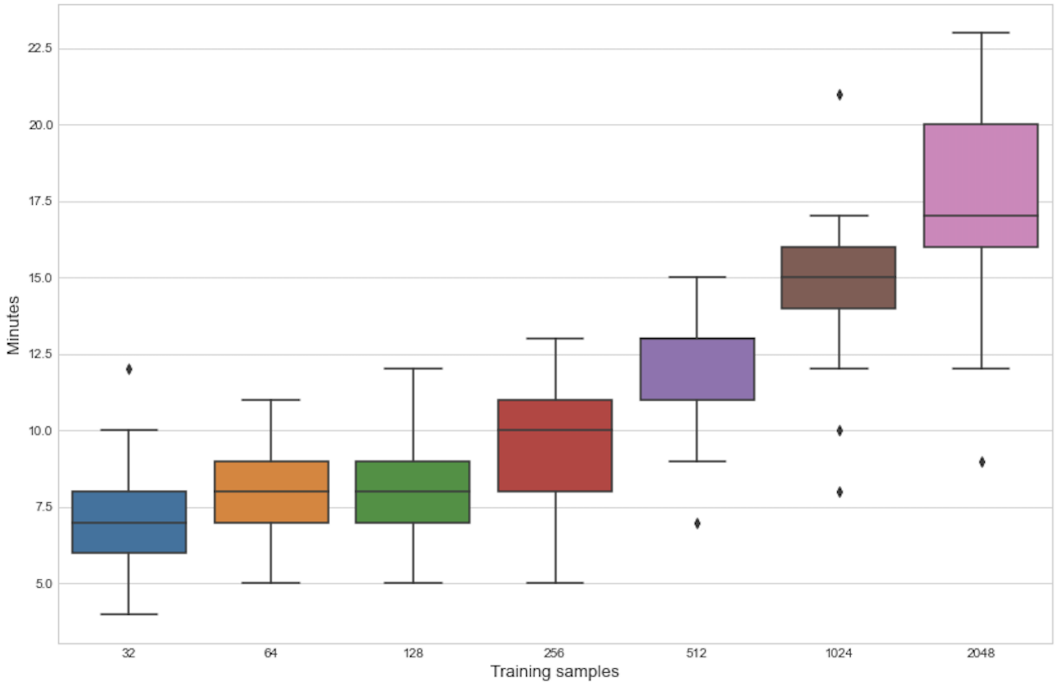


Figure 2: Boxplot showing the training time of the models on the `AngryTweets` dataset with varying number of training samples.

Lastly, Swedish does not have a NER tagged version of the corresponding dependency treebank, but they instead have the `SUC3` dataset, a NER-enriched version of the *Stockholm-Umeå Corpus* (Gustafson-Capková and Hartmann, 2006). This dataset does not follow the CONLL format and is instead released in the XML format, with the `<name>` XML tags containing the NER tags for

the words they span over [8]. This dataset contains the NER tags `animal`, `event`, `inst`, `myth`, `other`, `person`, `place`, `product` and `work`. These were converted to `MISC`, `MISC`, `ORG`, `MISC`, `MISC`, `PER`, `LOC`, `MISC` and `MISC`, respectively.

4.2 Sentiment Classification

We treat the sentiment classification task as a three-class classification task, with the classes `positive`, `neutral` and `negative`. Evaluation of the models is done using Matthew's Correlation Coefficient (Matthews, 1975) as the primary metric as well as reporting the macro-average F1-score as a secondary metric. We choose to use Matthew's Correlation Coefficient as the primary metric as it has been shown to be more reliable than the macro-average F1-score (Chicco and Jurman, 2020), while also being the standard metric used in the `GLUE` (Wang et al., 2018) and `SuperGLUE` (Sarlin et al., 2020) benchmarks.

For Danish we use the sentiment classification dataset `AngryTweets` (Pauli et al., 2021), which contains crowdsourced annotations of Danish tweets. To comply with Twitter's Terms of Use we have fully anonymised the tweets by replacing all user mentions with `@USER` and all links by `[LINK]`, as well as shuffling the tweets.

For Norwegian we included the sentiment classification dataset `NoReC` (Norwegian Review Corpus) (Velldal et al., 2018), which are based on scraped reviews from Norwegian websites.

Lastly, for Swedish we use the sentiment classification dataset presented in Svensson (2017), which is based on reviews from the Swedish websites `www.reco.se` and `se.trustpilot.com`. In analogy with `NoReC` we dub this dataset the Swedish Review Corpus (`SweReC`).

4.3 Linguistic Acceptability

Based on the inclusion of the CoLA (Corpus of Linguistic Acceptability) dataset (Warstadt et al., 2019) in the GLUE benchmark (Wang et al., 2018), we construct new linguistic acceptability datasets for the Scandinavian languages. This task is often framed as a binary classification task, where the model is tasked with predicting whether a given sentence is grammatically correct or not.

We dub our new datasets Scandinavian Linguistic Acceptability (`ScaLA`), which we release for Danish, Norwegian Bokmål, Norwegian Nynorsk, Swedish, Icelandic and Faroese. Each of these datasets consist of 1,024 training samples, 256 validation samples and 2,048 test samples, in accordance with Section 3.4. The `ScaLA` datasets are based on the Danish, Norwegian, Swedish, Icelandic and Faroese versions of the Universal Dependencies datasets (Kromann and Lynge, 2004; Øvrelid and Hohle, 2016; Nivre et al., 2006; Rögnvaldsson et al., 2012; Jónsdóttir and Ingason, 2020; Arnardóttir et al., 2020).

Firstly, we assume that the documents in the Universal Dependencies datasets are grammatically correct, an assumption we have been able to verify for the Danish part, by manually inspecting a random sample of the documents. We create negative examples by *either* removing a single word or swapping two consecutive words, where only one such "corruption" is applied to each negative sample.

Naively corrupting the documents in this way does not always lead to grammatically incorrect samples, however. For instance, removing the word "rød" (red) from the sentence "Den røde bil er stor" (The red car is big) does not lead to an incorrect sentence "Den bil er stor" (The car is big).

In order to ensure that the resulting sentence is indeed grammatically correct, we enforce restrictions on the words that can be removed or swapped. We have gone for a conservative approach, where we have systematically checked corruptions of words with a given part-of-speech tag, and only allow corruptions that were always grammatically correct in our tests. This led us to the following restrictions:

1. We do not remove adjectives, adverbs, punctuation, determiners or numbers, as the resulting sentence will still be grammatically correct in most cases.

2. We do not remove nouns or proper nouns if they have another noun or proper noun as neighbour, as again that usually does not make the sentence incorrect either.

3. When swapping two neighbouring words, we require them to have different POS tags.

4. We do not swap punctuation or symbols.

5. If we swap the first word then we ensure that the swapped words have correct casing.

[8] The `<ne>` XML tags are also NER tags, but these have been automatically produced by `SpaCy` (Honnibal et al., 2020) models and are thus not gold standard.

We are able to enforce these restrictions as we have gold-standard POS tokens available for these datasets.

4.4 Question Answering

We also construct new question answering datasets for the Mainland Scandinavian languages, as we are not aware of any existing datasets for these languages. We dub these datasets ScandiQA, which we release for each of the Mainland Scandinavian languages.

These datasets are based on the MKQA dataset (Longpre et al., 2021), which is based on the *Natural Questions* (NQ) dataset (Kwiatkowski et al., 2019). The NQ dataset contains questions inputted to Google's search engine, associated with the HTML page of the search result. In many cases these questions have an answer associated with it (a so-called *short answer*) which appears in the HTML, and in some cases they also have the paragraph in which the short answer appears (a so-called *long answer*).

The MKQA dataset contains human translations of 10,000 questions and short answers into 26 languages, including Danish, Norwegian and Swedish. Aside from adding these translations, the MKQA dataset also corrects many mistakes in the original NQ dataset by including answers not present in the original dataset, or by correcting the short answers chosen in the original dataset.

The main thing missing from the MKQA dataset is the context paragraph, which is what we add to the dataset as follows. For each MKQA sample, we first locate the corresponding sample in the NQ dataset. If that sample has a long answer then we use that as the initial (English) context. Otherwise, if neither the NQ dataset nor the MKQA dataset has an answer registered, then we use the paragraph in the HTML with the largest cosine similarity to the question, where we embed the documents using the Sentence Transformer (Reimers and Gurevych, 2019) model all-mpnet-base-v2.[9]

In the last case, where there is no long answer for the sample in NQ but there *is* an answer in MKQA, we want to identify the paragraph in the HTML containing the MKQA answer. Unfortunately, the MKQA answers do no appear verbatim in the HTML (for instance, all dates are standardised to the YYYY-MM-DD format). We thus start by forming a list of *answer candidates* based on the MKQA answer, which includes most of the ways dates and numerals are written in English. We then locate the paragraph containing any of the answer candidates and which has the largest cosine similarity to the question, where we embed the documents as described above.

The above procedure thus results in an English context paragraph containing the answer. We next translate this context paragraph to Danish and Swedish using the DeepL translation API[10]. As DeepL did not support Norwegian when we conducted this experiment, we translated the context paragraph to Norwegian using the Google Translation API[11] instead. With the contexts translated, we next extract all the answer candidates for the translated context relevant to the given Mainland Scandinavian language, and change the answer to the answer candidate appearing in the translated context. If no answer candidate appears in the translated context then we discard the sample.

The MKQA dataset also contains samples with *no* answer, and we include these samples in the ScandiQA dataset as well. For these samples, we simply use the translated context paragraphs as described above. The final dataset contains 7,810 Danish samples, 7,798 Swedish samples and 7,813 Norwegian samples. We release this dataset separately[12], as well as build a ScandEval version of it with the same train/dev/test size as the other ScandEval datasets. In the ScandEval version (with 1,024/256/2,048 train/val/test samples as stated in Section 3.4) we only include samples that contain an answer, as otherwise we found the 1,024 dataset size to be too small for this task.

We note that since this dataset is a translated version of a dataset originally written in English, it is not a perfect representation of the Mainland Scandinavian languages, as many of the questions and answers are concerned with topics specific to the USA. This might mean that pretrained multilingual models might have an advantage over monolingual models, but we leave this question for future work.

[9] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[10] https://www.deepl.com/pro-api

[11] https://cloud.google.com/translate/

[12] This can be found at https://huggingface.co/datasets/alexandrainst/scandi-qa and the source code is available at https://github.com/alexandrainst/ScandiQA.

4.5 Inference Speed

Aside from the predictive performance of the models we also benchmarked the inference speed of the finetuned models using the `pyinfer` package (Pierse, 2020), and report the mean number of inferences per second. This is done by recording the mean inference time of running a document with 2,600 characters[13] through the model one hundred times, and repeating that process 10 times. We also compute the confidence interval as described in Section 3.2. These have all been computed using an AMD Ryzen Threadripper 1920X 12-Core CPU.

5 Benchmarking Package and CLI

To enable every language researcher to benchmark their language models in a reproducible and consistent manner, we have developed a Python package called `scandeval`, which can benchmark any pretrained language model available on the Hugging Face Hub.

The `scandeval` package is implemented as both a CLI and a Python package, which enables ease of use as both a stand-alone benchmarking tool as well as enabling integration with other Python scripts. The package follows a very *opinionated* approach to benchmarking, meaning that very few parameters can be changed. This is a deliberate design decision to enable consistent benchmarking of all models. The package follows the hyperparameter choices described in Section 3.1. See more in the `scandeval` documentation.

6 Experiments

Using the `scandeval` package we have benchmarked more than 100 pretrained models in the Scandinavian languages which were available on the Hugging Face Hub. Aside from these models we also included several multilingual models to enable a fair comparison. Lastly, to enable better interpretability of the results, we also benchmark a randomly initialised XLM-RoBERTa-base model (Conneau et al., 2020) and an ELECTRA-small model (Clark et al., 2019) on the datasets, which will make it more transparent how much "external knowledge" the pretrained models are able to utilise in their predictions. Benchmarking all these models approximately required 1000 GPU hours on a GeForce RTX 2080 Ti GPU, which emitted approximately 40 kg of CO_2 equivalents[14].

6.1 Benchmarking Results

We have presented all of the benchmarked results along with their associated confidence intervals in an online leaderboard. These scores have been computed as described in Section 3, and the top-5 performing models for each language, as well as overall, can be found in Table 1.

We see from Table 1 that NB-BERT-large[15] (Kummervold et al., 2021) is the best performing model in Norwegian as well as overall, DFM-encoder-large-v1[16] being the best Danish model, and KB-BERT-large[17] (Malmsten et al., 2020) having the best performance in Swedish.

The massively multilingual models in the top 5 scores are RemBERT (Chung et al., 2020) and mDeBERTaV3 (He et al., 2021). The remaining models in the top 5 are NB-RoBERTa-base-scandi[18], DanskBERT (Snæbjarnarson et al., 2023), NB-BERT-base (Kummervold et al., 2021), NorBERT2 (Kutuzov et al., 2021), KB-BERT-base (Malmsten et al., 2020) and AI-Nordics-BERT-large[19].

6.2 Cross-lingual Transfer

This experiment investigated the cross-lingual transfer capabilities of the Scandinavian models, and tested our two hypotheses from Section 1. This used the methodology described in Section 3.3. For the Insular Scandinavian languages, the tasks included here are the Icelandic and Faroese versions of the `ScaLA` dataset, the Icelandic `NER` dataset `MIM-GOLD-NER` (Ingólfsdóttir et al., 2020) and the Faroese part of the `NER` dataset `WikiANN` (Rahimi et al., 2019). The resulting benchmark results can be found in Table 2 and all the raw scores can be found in the appendix. The results affirm our two hypotheses, as we see that the group

[13] The document is "This is a dummy document. ", repeated 100 times.

[14] With a power usage of 250 W/h (Techpowerup.com) and a carbon efficiency of 0.16 kg/kWh in Denmark (Ritchie et al., 2022).

[15] `https://huggingface.co/NbAiLab/nb-bert-large`

[16] `https://huggingface.co/chcaa/dfm-encoder-large-v1`

[17] `https://huggingface.co/KBLab/megatron-bert-large-swedish-cased-165k`

[18] `https://huggingface.co/NbAiLab/nb-roberta-base-scandi`

[19] `https://huggingface.co/AI-Nordics/bert-large-swedish-cased`

Rank	Overall	Danish	Norwegian	Swedish
1	NB-BERT-large	DFM-encoder-large-v1	NB-BERT-large	KB-BERT-large
2	DFM-encoder-large-v1	NB-BERT-large	NB-BERT-base	NB-BERT-large
3	RemBERT	DanskBERT	NB-RoBERTa-base-scandi	KB-BERT-base
4	mDeBERTaV3-base	RemBERT	NorBERT2	AI-Nordics-BERT-large
5	NB-RoBERTa-base-scandi	mDeBERTaV3-base	mDeBERTaV3	RemBERT

Table 1: The five best performing pretrained models in the Mainland Scandinavian language categories.

of languages with the largest F-statistic is the group of Mainland Scandinavian languages.

Benchmark group	F-statistic	Benchmark group	F-statistic
da, no	16.81	da, sv, is	4.36
da, sv	15.48	da, sv, fo	10.72
da, is	4.76	da, is, fo	5.48
da, fo	7.29	no, sv, is	3.11
no, sv	8.14	no, sv, fo	5.57
no, is	3.73	no, is, fo	3.64
no, fo	2.70	sv, is, fo	4.26
sv, is	4.48	da, no, sv, is	6.97
sv, fo	7.59	da, no, sv, fo	25.40
is, fo	21.84	da, no, is, fo	4.97
da, no, sv	**33.34**	da, sv, is, fo	5.21
da, no, is	4.27	no, sv, is, fo	3.38
da, no, fo	11.56	da, no, sv, is, fo	7.53

Table 2: F-statistics showing the cross-lingual transfer between the Scandinavian language models. Here da is Danish, no is Norwegian, sv is Swedish, is is Icelandic and fo is Faroese.

7 Discussion

We note that the benchmarking results presented in Section 6.1 show that the efforts of the National Libraries in Norway and Sweden, as well as the Danish Foundation Models project in Denmark, have paid off, in the sense that their models NB-BERT-large (Kummervold et al., 2021), KB-BERT-large (Malmsten et al., 2020) and DFM-encoder-large-v1 are outperforming the multilingual models.

This seems to indicate that investing in language technologies at a large language-specific level can be worthwhile. We also see from the same table that the Norwegian model is within the top two best models in Danish, Norwegian and Swedish, indicating a potentially large amount of language transfer, supported by the cross-lingual transfer experiment in Section 6.2. This indicates that a joint Mainland Scandinavian approach could improve the results of the current monolingual models within the Mainland Scandinavian languages.

8 Conclusion

In this paper we have presented a benchmarking framework for the Scandinavian languages, together with a Python package and CLI, scandeval, which can be used to benchmark any model available on the Hugging Face Hub. The benchmark features four tasks: named entity recognition, sentiment classification, linguistic acceptability and question answering. We have also released two new datasets, ScaLA and ScandiQA, which constitute the linguistic acceptability and question answering tasks, respectively. We have benchmarked more than 100 models on the Mainland Scandinavian datasets in the benchmark and presented these results in an online leaderboard. In our analysis of the benchmarking results we have shown substantial cross-lingual transfer between the Mainland Scandinavian languages, and no notable transfer between the group of Mainland Scandinavian languages and the group of Insular Scandinavian languages. This is the justification for including only the Mainland Scandinavian languages in the online leaderboard while maintaining support for the Insular Scandinavian languages in the scandeval package.

References

2021. *Oxford English Dictionary*. Oxford University Press. https://www.lexico.com/definition/scandinavia.

Amine Abdaoui, Camille Pradel, and Grégoire Sigel. 2020. Load What You Need: Smaller Versions of Mutililingual BERT. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 119–123, Online. Association for Computational Linguistics.

Yvonne Adesam, Aleksandrs Berdicevskis, and Felix Morger. 2020. SwedishGLUE – Towards a Swedish Test Set For Evaluating Natural Language Understanding Models. *Institutionen för svenska språket*.

Þórunn Arnardóttir, Hinrik Hafsteinsson, Einar Freyr Sigurðsson, Kristín Bjarnadóttir, Anton Karl Ingason, Hildur Jónsdóttir, and Steinþór Steingrímsson. 2020. A Universal Dependencies conversion pipeline for a Penn-format constituency treebank. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 16–25.

Lukasz Augustyniak, Kamil Tagowski, Albert Sawczyn, Denis Janiak, Roman Bartusiak, Adrian Dominik Szymczak, Arkadiusz Janz, Piotr Szymański, Marcin Wątroba, Mikołaj Morzy, et al. This is the way: designing and compiling LEPISZCZE, a comprehensive NLP benchmark for Polish. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Davide Chicco and Giuseppe Jurman. 2020. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21:1–13.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2020. Rethinking Embedding Coupling in Pre-trained Language Models. In *International Conference on Learning Representations*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Stefan Daniel Dumitrescu, Petru Rebeja, Beata Lorincz, Mihaela Gaman, Andrei Avram, Mihai Ilie, Andrei Pruteanu, Adriana Stan, Lorena Rosia, Cristina Iacobescu, et al. 2021. LiRo: Benchmark and leaderboard for romanian language tasks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Kenneth Enevoldsen, Lasse Hansen, and Kristoffer L Nielbo. 2021. DaCy: A Unified Framework for Danish NLP. *Proceedings http://ceur-ws. org ISSN*, 1613:0073.

Jan Terje Faarlund. 2019. *The syntax of mainland Scandinavian*. Oxford University Press.

Sofia Gustafson-Capková and Britt Hartmann. 2006. Manual of the Stockholm Umeå corpus version 2.0. *Stockholm University*. https://spraakbanken.gu.se/parole/Docs/SUC2.0-manual.pdf.

Jiyeon Ham, Yo Joong Choe, Kyubyong Park, Ilji Choi, and Hyungjoon Soh. 2020a. KorNLI and KorSTS: New benchmark datasets for Korean natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 422–430, Online. Association for Computational Linguistics.

Jiyeon Ham, Yo Joong Choe, Kyubyong Park, Ilji Choi, and Hyungjoon Soh. 2020b. Kornli and korsts: New benchmark datasets for korean natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 422–430.

Einar Haugen. 1976. *The Scandinavian languages: An introduction to their history*. Cambridge, Mass., Harvard University Press.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Matthew Honnibal, Ines Montani, Sofie Van Landeglem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Rasmus Hvingelby, Amalie Brogaard Pauli, Maria Barrett, Christina Rosted, Lasse Malm Lidegaard, and Anders Søgaard. 2020. DaNE: A named entity resource for Danish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4597–4604, Marseille, France. European Language Resources Association.

Malte Højmark-Bertelsen. 2021. Ælæctra - A Step Towards More Efficient Danish Natural Language Processing. https://github.com/MalteHB/-l-ctra/.

Svanhvít L Ingólfsdóttir, Ásmundur A Gudjónsson, and Hrafn Loftsson. 2020. Named Entity Recognition for Icelandic: Annotated Corpus and Models. In *International Conference on Statistical Language and Speech Processing*, pages 46–57. Springer.

Tim Isbister and Magnus Sahlgren. 2020. Why Not Simply Translate? A First Swedish Evaluation Benchmark for Semantic Similarity. *arXiv preprint arXiv:2009.03116*.

Hildur Jónsdóttir and Anton Karl Ingason. 2020. Creating a parallel Icelandic dependency treebank from raw text to Universal Dependencies. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2924–2931.

Fredrik Jørgensen, Tobias Aasmoe, Anne-Stine Ruud Husevåg, Lilja Øvrelid, and Erik Velldal. 2020. NorNE: Annotating named entities for Norwegian. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4547–4556, Marseille, France. European Language Resources Association.

Simran Khanuja, Melvin Johnson, and Partha Talukdar. 2021. MergeDistill: Merging language models using pre-trained distillation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2874–2887, Online. Association for Computational Linguistics.

Daniel Khashabi, Arman Cohan, Siamak Shakeri, Pedram Hosseini, Pouya Pezeshkpour, Malihe Alikhani, Moin Aminnaseri, Marzieh Bitaab, Faeze Brahman, Sarik Ghazarian, Mozhdeh Gheini, Arman Kabiri, Rabeeh Karimi Mahabadi, Omid Memarrast, Ahmadreza Mosallanezhad, Erfan Noury, Shahab Raji, Mohammad Sadegh Rasooli, Sepideh Sadeghi, Erfan Sadeqi Azer, Niloofar Safi Samghabadi, Mahsa Shafaei, Saber Sheybani, Ali Tazarv, and Yadollah Yaghoobzadeh. 2020. ParsiNLU: A Suite of Language Understanding Challenges for Persian. *Transactions of the Association for Computational Linguistics*, 9:1147–1162.

Fajri Koto, Afshin Rahimi, Jey Han Lau, and Timothy Baldwin. 2020. IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 757–770, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Matthias Trautner Kromann and Stine Kern Lynge. 2004. The Danish Dependency Treebank v. 1.0.

Per E Kummervold, Javier De la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. 2021. Operationalizing a National Digital Library: The Case for a Norwegian Transformer Model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 20–29, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-scale contextualised language modelling for norwegian. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 30–40.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the national library of sweden–making a swedish bert. *arXiv preprint arXiv:2007.01658*.

Brian W Matthews. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

Jens Dahl Møllerhøj. 2020. Nordic BERT. `https://github.com/certainlyio/nordic_bert`.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Lilja Øvrelid and Petter Hohle. 2016. Universal Dependencies for Norwegian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1579–1585, Portorož, Slovenia. European Language Resources Association (ELRA).

Amalie Brogaard Pauli, Maria Barrett, Ophélie Lacroix, and Rasmus Hvingelby. 2021. DaNLP: An open-source toolkit for Danish natural language processing. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 460–466, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Charles Pierse. 2020. Pyinfer. `https://github.com/cdpierse/pyinfer`.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

David C Plaut, Steven J Nowlan, and Geoffrey E Hinton. 1986. Experiments on learning by back propagation. *Technical Report CMU-CS-86-126*.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Hannah Ritchie, Max Roser, and Pablo Rosado. 2022. Energy. *Our World in Data*. Https://ourworldindata.org/energy.

Eiríkur Rögnvaldsson, Anton Karl Ingason, Einar Freyr Sigurðsson, and Joel Wallenberg. 2012. The Icelandic parsed historical corpus (IcePaHC). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1977–1984.

Piotr Rybak, Robert Mroczkowski, Janusz Tracz, and Ireneusz Gawlik. 2020. KLEJ: Comprehensive benchmark for Polish language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1191–1201, Online. Association for Computational Linguistics.

Ali Safaya, Emirhan Kurtuluş, Arda Goktogan, and Deniz Yuret. 2022. Mukayese: Turkish nlp strikes back. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 846–863.

Magnus Sahlgren, Fredrik Carlsson, Fredrik Olsson, and Love Börjeson. 2021. It's basically the same language anyway: the case for a nordic language model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 367–372, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947.

Philip Tamimi Sarnikowski. 2021. Danish Transformers. GitHub. `https://github.com/sarnikowski`.

Tatiana Shavrina, Alena Fenogenova, Emelyanov Anton, Denis Shevelev, Ekaterina Artemova, Valentin Malykh, Vladislav Mikhailov, Maria Tikhonova, Andrey Chertok, and Andrey Evlampiev. 2020. RussianSuperGLUE: A Russian language understanding evaluation benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4717–4726, Online. Association for Computational Linguistics.

Vésteinn Snæbjarnarson, Annika Simonsen, Goran Glavaš, and Ivan Vulić. 2023. Transfer to a low-resource language via close relatives: The case study on faroese. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, Tórshavn, Faroe Islands. Linköping University Electronic Press, Sweden.

Kristoffer Svensson. 2017. Sentiment Analysis With Convolutional Neural Networks: Classifying sentiment in Swedish reviews. Bachelor's thesis.

Techpowerup.com. NVIDIA GeForce RTX 2080 Ti Specs. `https://www.techpowerup.com/gpu-specs/geforce-rtx-2080-ti.c3305`. Accessed: 2023-04-01.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Erik Velldal, Lilja Øvrelid, Eivind Alexander Bergem, Cathrine Stadsnes, Samia Touileb, and Fredrik Jørgensen. 2018. NoReC: The Norwegian review corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel Bowman. 2019. Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiaohong Li, Zhi Yuan Lim, Sidik Soleman, Rahmad Mahendra, Pascale Fung, Syafri Bahar, and Ayu Purwarianti. 2020. IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 843–857, Suzhou, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Beilei Xiang, Changbing Yang, Yu Li, Alex Warstadt, and Katharina Kann. 2021. CLiMP: A benchmark for Chinese language model evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2784–2790, Online. Association for Computational Linguistics.

Liang Xu, Xuanwei Zhang, Lu Li, Hai Hu, Chenjie Cao, Weitang Liu, Junyi Li, Yudong Li, Kai Sun, Yechen Xu, Yiming Cui, Cong Yu, Qianqian Dong, Yin Tian, Dian Yu, Bo Shi, Jun jie Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhen-Yi Yang, Kyle Richardson, and Zhenzhong Lan. 2020. CLUE: A Chinese Language Understanding Evaluation Benchmark. In *International Conference on Computational Linguistics*.

A Cross-lingual transfer experiment

Language	Hugging Face Model ID	# Parameters
Danish	`vesteinn/DanskBERT`	124M
Swedish	`KB/bert-base-swedish-cased`	125M
Norwegian	`patrickvonplaten/norwegian-roberta-base`	125M
Icelandic	`mideind/IceBERT`	124M
English	`roberta-base`	125M
German	`deepset/gbert-base`	110M
Dutch	`pdelobelle/robbert-v2-dutch-base`	117M
Finnish	`TurkuNLP/bert-base-finnish-cased-v1`	125M
Russian	`DeepPavlov/rubert-base-cased`	178M
Arabic	`asafaya/bert-base-arabic`	111M

Table 3: The Hugging Face Hub model IDs of the models used in the cross-lingual transfer experiment.

Model	Danish Score	Norwegian Score	Swedish Score	Icelandic Score	Faroese Score
`Danish`	**63.87 ± 1.26**	53.74 ± 3.73	52.08 ± 2.70	30.39 ± 1.55	45.26 ± 1.35
`Norwegian`	46.30 ± 2.83	**58.78 ± 1.44**	46.90 ± 2.79	28.85 ± 1.45	43.35 ± 2.20
`Swedish`	45.81 ± 2.96	47.32 ± 2.66	**69.29 ± 1.40**	28.69 ± 1.61	43.63 ± 1.90
`Icelandic`	30.20 ± 1.23	28.68 ± 2.91	36.80 ± 2.14	**71.00 ± 1.50**	**48.26 ± 4.76**
`Finnish`	32.55 ± 1.47	30.71 ± 2.14	38.94 ± 1.51	16.33 ± 1.89	36.87 ± 1.17
`English`	34.11 ± 2.11	30.92 ± 2.69	39.24 ± 1.92	28.39 ± 2.41	40.75 ± 1.59
`German`	28.13 ± 2.04	27.58 ± 2.90	37.62 ± 4.18	26.13 ± 1.63	41.02 ± 1.46
`Dutch`	31.78 ± 1.62	28.27 ± 2.51	35.06 ± 1.87	26.21 ± 1.79	40.83 ± 1.70
`Russian`	33.91 ± 1.88	33.55 ± 2.17	39.14 ± 2.33	29.96 ± 1.58	43.17 ± 1.66
`Arabic`	22.89 ± 1.82	19.98 ± 2.24	25.40 ± 2.69	10.33 ± 2.19	35.33 ± 1.57

Table 4: The raw benchmarking results used in the cross-lingual transfer experiment.

B Training Data Size Experiment

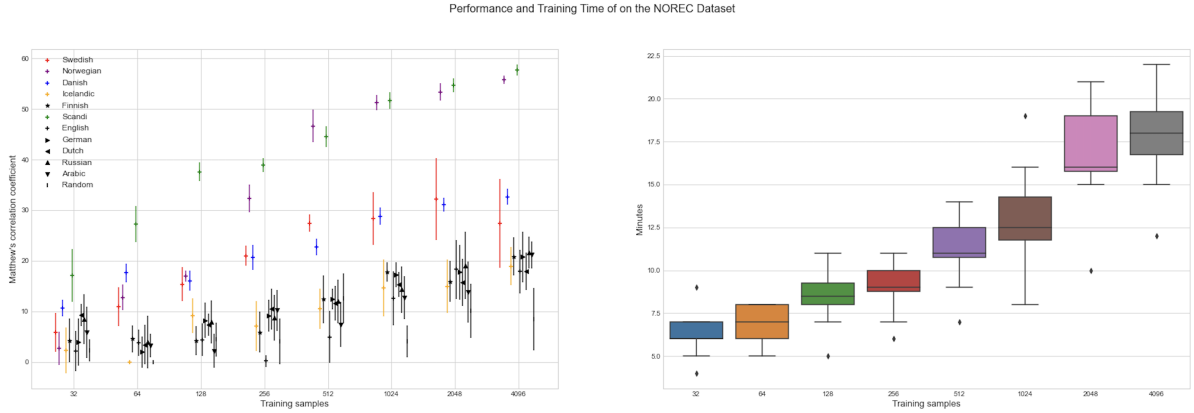


Figure 3: The results from the training data size experiment for the `NoReC` dataset.

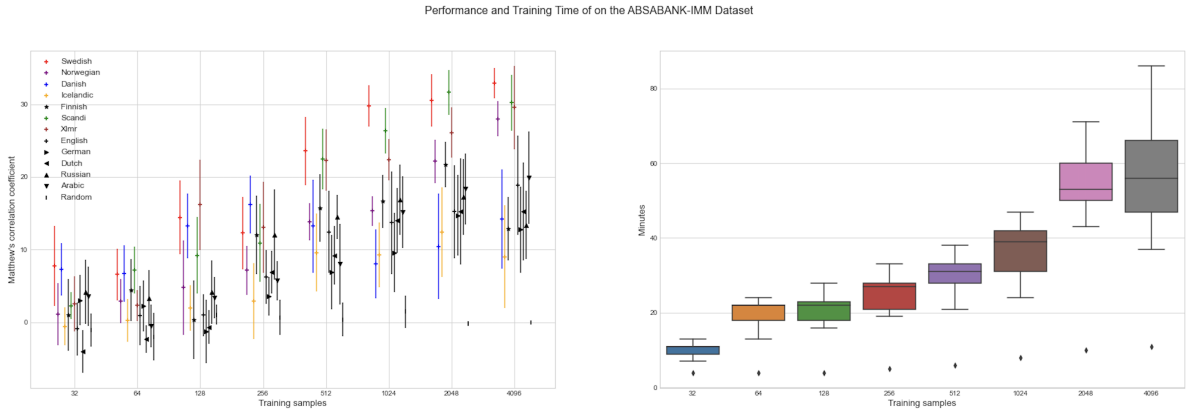


Figure 4: The results from the training data size experiment for the `Absabank-Imm` dataset (Adesam et al., 2020).

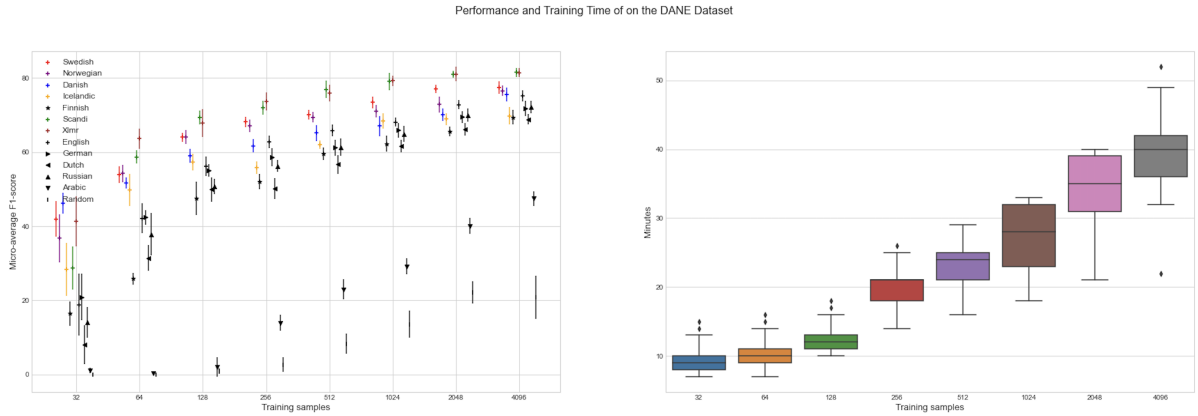


Figure 5: The results from the training data size experiment for the `DaNE` dataset.

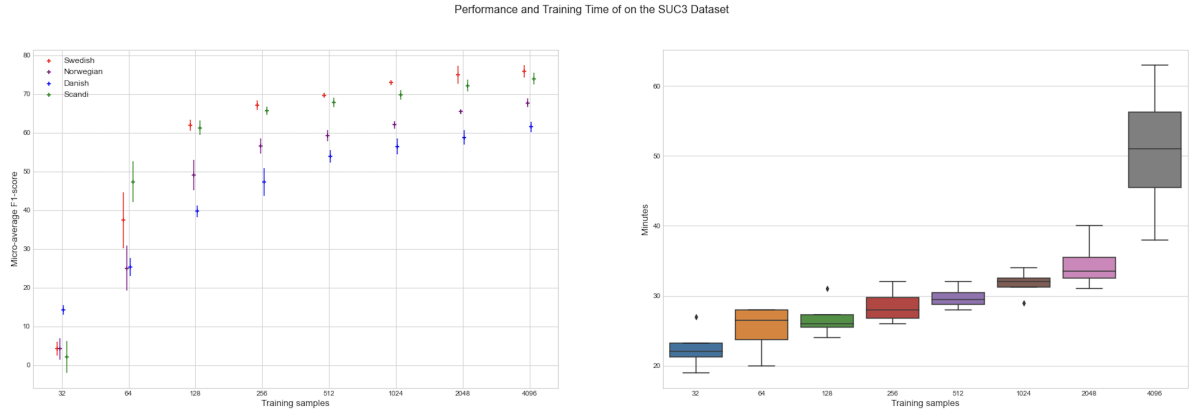


Figure 6: The results from the training data size experiment for the SUC3 dataset.

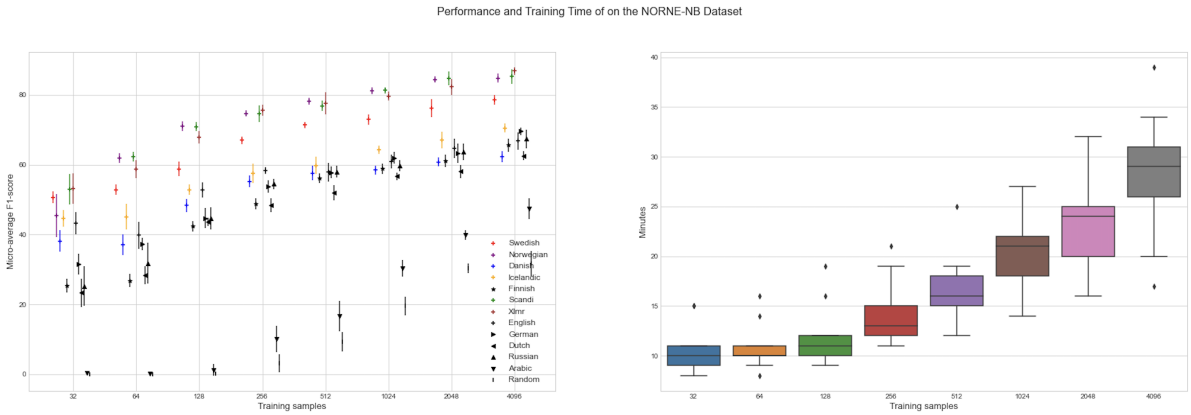


Figure 7: The results from the training data size experiment for the NorNE-NB dataset.

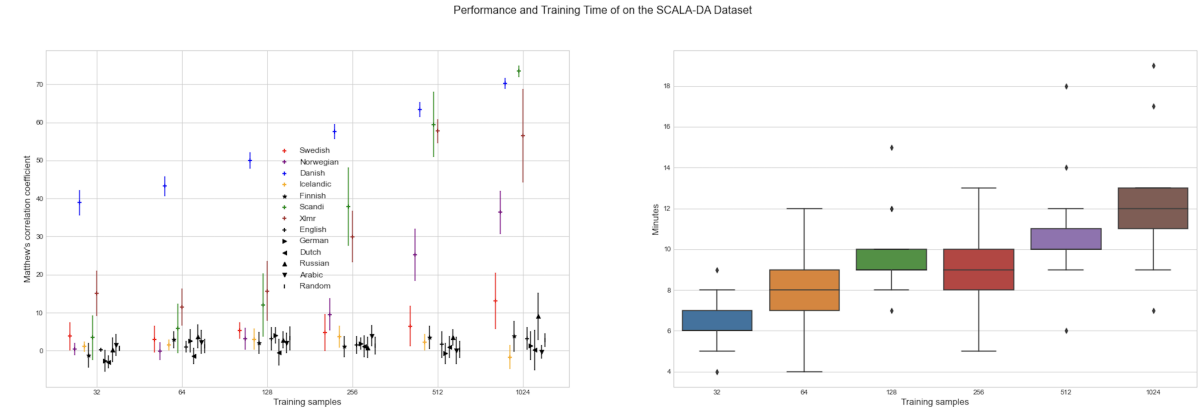


Figure 8: The results from the training data size experiment for the ScaLA-DA dataset.

Figure 9: The results from the training data size experiment for the `ScaLA-SV` dataset.

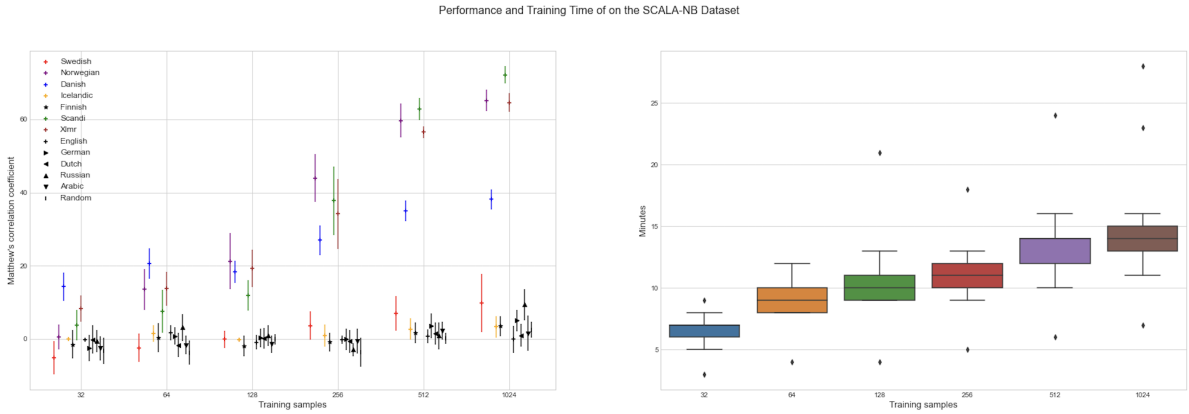


Figure 10: The results from the training data size experiment for the `ScaLA-NB` dataset.

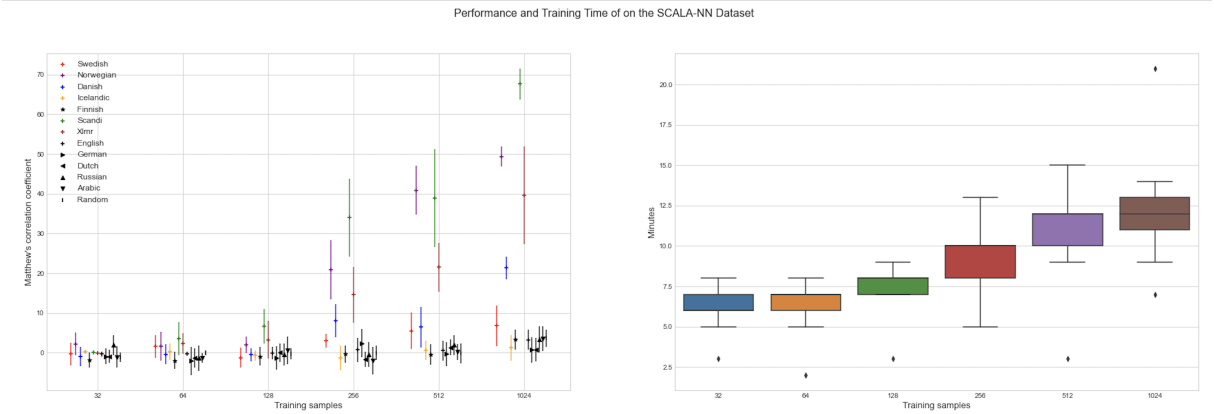


Figure 11: The results from the training data size experiment for the `ScaLA-NN` dataset.

BRENT: Bidirectional Retrieval Enhanced Norwegian Transformer

Lucas Georges Gabriel Charpentier,* Sondre Wold,*
David Samuel and Egil Rønningstad

University of Oslo, Language Technology Group
{lgcharpe|sondrewo|egilron|davisamu}@ifi.uio.no

Abstract

Retrieval-based language models are increasingly employed in question-answering tasks. These models search in a corpus of documents for relevant information instead of having all factual knowledge stored in its parameters, thereby enhancing efficiency, transparency, and adaptability. We develop the first Norwegian retrieval-based model by adapting the REALM framework and evaluate it on various tasks. After training, we also separate the language model, which we call the *reader*, from the retriever components, and show that this can be fine-tuned on a range of downstream tasks. Results show that retrieval augmented language modeling improves the reader's performance on extractive question-answering, suggesting that this type of training improves language models' general ability to use context and that this does not happen at the expense of other abilities such as part-of-speech tagging, dependency parsing, named entity recognition, and lemmatization. Code, trained models, and data are made publicly available.[1]

1 Introduction

Retrieval-based language models meet some important shortcomings associated with pre-trained language models (PLMs): they are more dynamic, allowing for updating of knowledge without having to re-train the model from scratch; they are more transparent, allowing backtracking the source of returned statements; and they are more efficient, as retrieval provides a non-parametric memory. The accentuated benefit of these models has been the

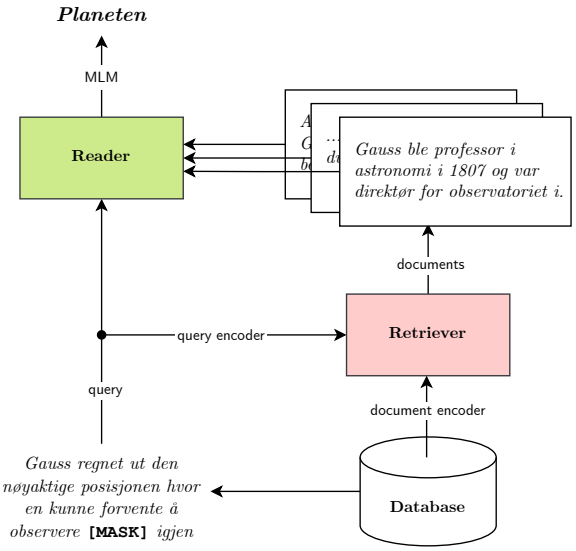


Figure 1: The proposed architecture, based on the REALM method from Guu et al. (2020).

OpenQA task – where they have established new state-of-the-art results on datasets like NaturalQuestions (Kwiatkowski et al., 2019) and WebQuestions (Berant et al., 2013). There, models first fetch a relevant passage from a data source in order to be able to answer a question — as compared to extractive QA, where a passage with the correct answer is provided explicitly as additional input to the model, also referred to as machine reading comprehension.

In this work, we develop the first Norwegian retrieval-based model, BRENT: **B**idirectional **R**etrieval **E**nhanced **N**orwegian **T**ransformer, based on the general approach proposed by Guu et al. (2020). Our model consists of two *encoders* that respectively learn to embed documents and queries into dense vector representations, and a *reader* module that learns to utilize the retrieved context for prediction, as shown in Figure 1. These are trained jointly and end-to-end, and we start their training from an already pre-trained Norwegian LM. Compared to previous work, we use a

*The authors contributed equally to this work
[1] https://github.com/ltgoslo/brent

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 202–214
May 22-24, 2023

relatively small retrieval corpus consisting of 730k Wikipedia documents.

The learning objective is masked language modeling (MLM), and the top k most relevant documents are retrieved from the retrieval corpus through a maximum inner product search (MIPS).

The size of our retrieval corpus allows us to update the search index synchronously during training and do exact matching, as opposed to the asynchronous updates and approximations done in Guu et al. (2020). Furthermore, we do not consider OpenQA as an evaluation task, but instead, we study how retrieval-augmented language modeling can be used as a continued pre-training step in order to improve context utilization in the reader model. That is, we evaluate the reader as a stand-alone — extracting it from the overall pipeline so that it can be distributed and used as a normal LM.

In order to analyze the effect of this continued pre-training, we also benchmark the reader against other NLP tasks that by intuition should not benefit from this type of training, such as part-of-speech-tagging, named entity recognition, dependency parsing, and lemmatization. We find that the retrieval-augmented training procedure increases the reader's performance on extractive QA without decreasing performance on these tasks. However, we find that it decreases performance on both targeted and sentence-level sentiment analysis. To summarize, our contributions are:

- We develop and release the first Norwegian retrieval-based language model.
- We study how retrieval improves the reader's ability to use context on extractive QA while still performing on par with comparable baselines on morpho-syntactic tasks.
- We analyze the different components of a retrieval-based system through a series of ablations, addressing problems associated with common design choices.

2 Related work

The basic setup for most retrieval-based approaches to NLP is that for a query q, be it a question for QA or a premise in natural language inference, the model must retrieve a set of passages relevant to q. Relevant candidates are then typically appended to q before being passed to a classification layer.

While earlier work approached this using heuristics and sparse retrieval methods like BM25 (Robertson et al., 2009), recent work has focused on learning this retrieval step. Most of these use an architecture with an *encoder* and a *reader*: the encoder learns to represent q and the retrieved passages in a representation space that makes it possible to match documents using the inner product operation, while the reader learns how to utilize the retrieved passage for downstream prediction. Recent work by Jiang et al. (2022) shows how it is also possible to model this interaction using a single transformer model (Vaswani et al., 2017) with a *retrieval as attention* technique, as compared to having separate encoders and readers.

Lee et al. (2019) note that it is computationally impractical to learn to make predictions conditioned on a retrieval corpus from scratch and thus proposed to pre-train the encoders with an inverse cloze task (ICT) in order to "prime" the model for retrieval. This is also done in Sachan et al. (2021). We outline more details on this and how we use ICT in the following section.

The most direct application of retrieval is to use a supervision signal such as OpenQA to train the context and passage encoders, such as in Khattab et al. (2021). However, Guu et al. (2020) show how this setup can also be used for language modeling. Using English Wikipedia as the retrieval corpus, they perform MLM conditioned on retrieved passages. Passages are retrieved using MIPS over an index that is asynchronously updated during training. They also use a masking technique that prioritizes named entities in order to incentivize the usage of world knowledge from the retrieved passages. A similar approach to language modeling is also done in Borgeaud et al. (2022), but over a corpus consisting of trillions of tokens. For both works, the LMs are trained for a number of steps with retrieval before being fine-tuned on a downstream task, as is the typical workflow with PLMs.

Lewis et al. (2020b) demonstrate how the encoder-reader architecture can be used for language generation as well. They propose both a sequence model where the generation is conditioned on the same set of retrieved documents for the entire sequence and a token model where a different document is used per target token. The retriever is based on Dense Passage Retrieval (DPR) (Karpukhin et al., 2020), which uses the same general approach to retrieval as Guu et al. (2020), where a PLM like BERT (Devlin et al., 2019) is

used as the encoder. The reader model is swapped with a generator based on BART (Lewis et al., 2020a).

3 Method

As in Guu et al. (2020), the architecture of BRENT can be separated into two parts: a retriever and a reader. Our architecture is modified to improve the speed of training, to ensure that the retrieved documents affect the predictions, and to incentivize the retrieval of world knowledge from the retrieval corpus instead of the reader memorizing it. This section puts forth the architecture and these modifications.

3.1 Architecture

Retriever The first part of BRENT is the retriever, which consists of two components: the Query Encoder ($\text{Enc}_{\text{query}}$) and the Document Encoder (Enc_{doc}). Both have their own sets of weights and in our case have a BERT-style architecture and tokenizer. However, these can be initialized with other types of dense representation learners and could potentially also share weights for faster training.

The retriever receives as input the query, q, which is the masked sentence from the pre-training corpus, and passes it to $\text{Enc}_{\text{query}}$ to get a dense representation. Enc_{doc} encodes all the documents in the retrieval corpus. Once all the documents and the query are encoded, a similarity score is calculated between each document d and the query q:

$$\text{sim}(q, d) = \frac{\text{Enc}_{\text{query}}(q)^T \text{Enc}_{\text{doc}}(d)}{\sqrt{h_{\text{dim}}}},$$

where h_{dim} represents the encoding dimension of the retriever encoders. Since the query and doc vectors are not normalized, the inner product can be very large. In order to stabilize the training, we scale the inner product by dividing it by the square root of the hidden dimension.

Once all the similarity scores are calculated, we use softmax to create a probability distribution over all the documents for a given query:

$$p(d|q) = \frac{\exp(\text{sim}(q, d))}{\sum_{d' \in D} \exp(\text{sim}(q, d'))}.$$

Finally, we create the inputs to the reader by appending the representations of each d to q, i.e. $[q; d]$. In other words, if we have a retrieval corpus D with N documents, then a single query generates N inputs to the reader — effectively multiplying by N the batch size passed to the reader. However, it is unfeasible to do this for the whole corpus, therefore we only retrieve the top-k documents based on the similarity scores.

Reader The reader is a single pre-trained language model taking as input the document d appended to query q ($[q; d]$). During continued pre-training, the reader is optimized for MLM. For each input to the model, we get predictions on what the masked words in the query are — given the context provided by document d. Formally, each input generates the following probability for the correct masked words y:

$$p(y|d, q) = \prod_{y_i \in M_q} p(y_i|d, q),$$

where y_i is the i-th masked word in query q and M_q is the set of all masked words in q. However, we want $p(y|q)$. Therefore, for a query q, we need to do k forward passes to get all the $p(y|d, q)$. Finally, to obtain $p(y|q)$ we marginalize:

$$p(y|q) = \sum_{d \in D} p(y|d, q)\, p(d|q).$$

Loss With $p(y|q)$ we can calculate the loss. During the loss function (cross-entropy) derivation, the error backpropagation is spread between the reader and the retriever. For the reader, this is the same as for any transformer-based model trained on the MLM task except that it averages over the batch size, number of retrieved documents, and the number of masked tokens. For the retriever, it is updated based on whether $p(y|d, q)$ was better or worse than $p(y|q)$. Specifically, if $p(y|d, q)$ is higher than $p(y|q)$, then the similarity score between q and d should increase. This can be seen with the following equation:

$$\nabla_\theta \log p(y|q) = \sum_{d \in D} u(d) p(d|q) \nabla_\theta \text{sim}(d, q)$$
$$u(d) = \left(\frac{p(y|d, q)}{p(y|q)} - 1 \right),$$

where θ represents the parameters of Enc_{doc}.[2] The same derivation applies to the parameters of $\text{Enc}_{\text{query}}$.

[2]The full derivation can be found in the appendix of Guu et al. (2020), where $z = d$, $x = q$ and f represents the function sim

3.2 Null Document

As pointed out in Guu et al. (2020), not all masked words need world knowledge to be predicted correctly. Therefore, we also add a null document appended to the query q. There are two ways to encode the null document. The first, and most obvious, is to pass the empty string to Enc_{doc} and use the resulting encoding as the null document. However, we use a parameter tensor initialized with all zeros instead. This saves us one forward and backward pass of Enc_{doc} per step, without affecting performance.

3.3 Corpus

We use a snapshot of the Norwegian Wikipedia from October 2022 as our corpus, limited to the Bokmål written standard. We pre-process the articles into chunks of token length 128, padding the last chunk of each article so that no chunk contains text from two different sources. After processing, the corpus consists of 735 000 documents, with an average number of words per chunk being about 100 ($\mu = 102, \sigma = 25$). We use this corpus for sampling sentences to mask for MLM and as a retrieval corpus during continued pre-training.

3.4 Search index

As described in our architecture, we use the documents d in the retrieval corpus D to improve the model's predictions. Ideally, we would use all the documents of the retrieval corpus to make the prediction. Then, the model would assign close to zero probabilities to most documents, while simultaneously having access to all documents, and therefore identifying the most relevant ones. However, this is not feasible, as it would require a very high amount of resources (which would keep increasing as we increase our retrieval corpus) and be unreasonably time-consuming. Therefore, we instead only retrieve the top-k documents in terms of similarity score. To be able to efficiently retrieve and find these documents, we use a search index. We build this index using the encoding of the documents produced by Enc_{doc}. Since we update Enc_{doc} at every backward pass, it follows that we should re-index the documents at each backward pass. However, this is too time-consuming and we therefore only re-index each s steps. We want to note here that there has been recent work on how to more efficiently retrieve from such an index (Alon et al., 2022; He et al., 2021). Since we use the same corpus for both MLM training and retrieval, the first retrieved document is often the same document from which the query comes from, as this will naturally have a high similarity score. To avoid directly giving our model the answer with the unmasked token in it, we make sure to remove this document.

3.5 Inverse cloze task

We warm up the encoders for both the query and the document with the ICT task from Lee et al. (2019) on $68k$ Wikipedia article introductions limited to 128 tokens, from a snapshot from October 2020. For each pass, the model must predict the relevant pseudo-document for a pseudo-question from a set of distractors. The question is a random sentence and the document is the text surrounding it, the distractors are sampled from the same batch.

3.6 Span Masking

For the MLM task, we combine both salient masking (Guu et al., 2020), where only named entities and dates that require world knowledge are masked, and random masking. We identify entities using an off-the-shelf named entity recognizer and dates with a simple parsing algorithm.[3] We use 15% salient masking, making sure to mask at least one salient span for each sample, and 3.75% random span masking (which is 25% of 15%). By doing this, we encourage the network to learn to retrieve spans requiring world knowledge while ensuring that the model is still able to model linguistic features.

4 Experiments

We evaluate BRENT on a wide range of Norwegian NLP tasks. We do this both without retrieval using the extracted reader, and with retrieval turned on using the full model. By doing this, we highlight both the improved capacity of the reader to use context and show how retrieval in general affects performance on NLP tasks other than QA. This section describes the specific datasets and models we use during experimentation.

4.1 Models

NorBERT2 A baseline Norwegian LM, originating from Kutuzov et al. (2021).

NorBERT2$_{50k}$ A NorBERT2 model trained for 50k additional steps on Wikipedia using MLM

[3] spaCy: https://spacy.io/

as described in Section 3.6, with a batch size of 1024. We show the performance of this model in order to get a more fair comparison, showcasing the improvements gained from the actual retrieval-augmented pre-training as compared to just doing regular pre-training for 50k more steps on the same corpora.

BRENT The entire model with retrieval turned on during fine-tuning. This is akin to a Norwegian version of REALM (Guu et al., 2020), but with our modifications. When subscripted, this indicates the source of the retrieval corpus, which could be either from Wikipedia or a task-specific dataset.

BRENT$_{\text{reader}}$ The reader model extracted after continued pre-training, used without any retrieval during fine-tuning on the downstream tasks.

4.2 Datasets

NorQuAD A Norwegian question answering dataset for machine reading comprehension (Ivanova et al., 2023) based on the SQuAD format (Rajpurkar et al., 2016). For a given question, the model must predict the correct span in a provided passage that answers the question. NorQuAD includes three domain splits: one sourced from the Norwegian Wikipedia ($N = 2351$), one from Norwegian news articles ($N = 2398$), and one split that combines both of them ($N = 4749$). We use an $80 - 10 - 10$ split on all three domains for training, validation, and testing.

NoReC$_{\textit{fine}}$ A fine-grained sentiment analysis dataset for Norwegian (Øvrelid et al., 2020). The texts are a subset of the NoReC dataset (Velldal et al., 2018), a multi-domain dataset of full-text professional reviews published in Norwegian online news sources. Each sentence in NoReC$_{\textit{fine}}$ is annotated for sentiment holders, targets, polar expressions, expression polarities, and polar intensities. A version for targeted sentiment analysis (TSA) is released on GitHub where only the sentiment targets are labeled.[4]

NoReC$_{\textit{sent}}$ A sentence-level sentiment analysis dataset for Norwegian derived from NoReC$_{\textit{fine}}$ (Øvrelid et al., 2020; Kutuzov et al., 2021). This dataset is generated by aggregating the entity sentiments in each sentence. The sentences are then labeled as either positive, negative, or neutral. We use the version only containing positive and negative sentiments. Both versions of the dataset (with and without neutral sentiment sentences) are available on GitHub.[5]

Morpho-syntactic tasks This group of tasks is based on annotations from the Norwegian Dependency Treebank (Solberg et al., 2014), which were converted to the Universal Dependencies (UD) format by Øvrelid and Hohle (2016) and later enriched with named-entity types by Jørgensen et al. (2020). The resulting dataset is called NorNE and we use its latest version.[6] The source of NorNE is mostly news texts, but also government reports, parliament transcripts, and blogs. We evaluate the models on all available UD tasks for Norwegian Bokmål (UPOS and UFeats tagging, lemmatization, and dependency parsing; Nivre et al., 2016),[7] as well as on named entity recognition (NER).[8]

4.3 Implementation details

Since running these models is resource intensive, we do not do a hyperparameter search. Instead, we base our hyperparameters on previous research where available. The following paragraphs outline the details of our experiments.

Search Index We use the FlatIndexIP from the FAISS (Johnson et al., 2019) library to construct our index. This allows us to get the most relevant documents rather than an approximation of the best documents. We can do this since our corpus of documents is relatively small. We retrieve the top-7 documents and append the null document, in essence retrieving 8 documents in total. We re-index every 100 steps.

ICT We use NorBERT2 as the initialization for the ICT warmup. For this, we use a learning rate of $1 * 10^{-4}$ and batch size of 128 for 10 epochs with early stopping on a single NVIDIA A100 GPU. After the warmup, these weights are then used as the starting point for $\text{Enc}_{\text{query}}$ and Enc_{doc} in the

[4] https://github.com/ltgoslo/norec_tsa

[5] https://github.com/ltgoslo/norec_sentence/

[6] https://github.com/ltgoslo/norne

[7] We use the official evaluation script from CoNLL 2018 shared task (Zeman et al., 2018, https://universaldependencies.org/conll18/evaluation.html).

[8] We employ the evaluation method from SemEval 2013 task 9.1 (Segura-Bedmar et al., 2013), re-implementated in https://github.com/davidsbatista/NER-Evaluation.

Model	Wiki		News		All	
	EM	F1	EM	F1	EM	F1
Human*	72.65	88.84	83.61	93.43	78.13	91.14
NorBERT2	$57.76^{\pm 1.15}$	$71.89^{\pm 0.89}$	$64.05^{\pm 1.27}$	$76.93^{\pm 1.15}$	$64.64^{\pm 1.40}$	$77.86^{\pm 0.65}$
$\text{NorBERT2}_{\text{50k}}$	$59.14^{\pm 0.55}$	$73.98^{\pm 1.05}$	$64.89^{\pm 1.44}$	$77.22^{\pm 0.57}$	$63.88^{\pm 0.49}$	$77.05^{\pm 0.55}$
$\text{BRENT}_{\text{reader}}$	$\mathbf{62.57}^{\pm 1.77}$	$\mathbf{76.45}^{\pm 1.40}$	$\mathbf{68.10}^{\pm 2.87}$	$\mathbf{80.40}^{\pm 1.71}$	$\mathbf{66.56}^{\pm 1.36}$	$\mathbf{80.01}^{\pm 1.16}$

Table 1: Results on different domains of the NorQuAD dataset. Results are reported as the mean and standard deviation over five random seeds. *Human performance is the mean performance of two annotators as reported in Ivanova et al. (2023))

Model	UPOS	UFeats	Lemma	LAS	NER
NorBERT2	$\mathbf{98.65}^{\pm 0.04}$	$\mathbf{97.58}^{\pm 0.06}$	$\mathbf{98.18}^{\pm 0.03}$	$\mathbf{93.15}^{\pm 0.05}$	$88.13^{\pm 0.34}$
$\text{NorBERT2}_{\text{50k}}$	$98.64^{\pm 0.04}$	$97.54^{\pm 0.04}$	$98.12^{\pm 0.06}$	$93.10^{\pm 0.22}$	$\mathbf{88.41}^{\pm 0.45}$
$\text{BRENT}_{\text{reader}}$	$98.62^{\pm 0.06}$	$97.55^{\pm 0.02}$	$98.09^{\pm 0.04}$	$92.96^{\pm 0.15}$	$87.70^{\pm 0.49}$

Table 2: Results on the morpho-syntactic tasks: accuracy of UPOS and UFeats tagging, the accuracy of lemmatization, the labeled attachment scores of dependency parsing, F1 scores of named entity recognition, where the evaluation requires an exact match on both span and label. Results are reported as the mean and standard deviation over five random seeds.

retriever, while the reader uses NorBERT2 without any warmup.

Pre-training We then train BRENT for 50k steps with a batch size of 1024 divided over 128 AMD MI250X GPUs,[9] a learning rate of $2 * 10^{-5}$, using the AdamW optimizer, and a Cosine scheduler with a warmup, on the chunked Wikipedia corpus. A full description of the model and the hyperparameters can be found in Appendix A.2.

Fine-tuning We run all experiments using five different seeds and report the average result and standard deviation. For the fine-tuning of the retrieval-enhanced models, we test both with and without re-indexing, i.e., fine-tuning Enc_{doc}. In both cases, we continue to fine-tune $\text{Enc}_{\text{query}}$. When fine-tuning, we use a higher learning rate for the retriever as compared to the reader, since we saw experimentally that this obtained better results. When re-indexing, we do it every 100 steps and at the end of each epoch. Hyperparameters for all evaluation tasks can be found in Appendix A.3. We fine-tune all models on a single GPU.

4.4 Results

4.4.1 Extractive QA

Table 1 shows the exact match (EM) and token-level F1 scores of different approaches on the NorQuAD dataset. $\text{BRENT}_{\text{reader}}$ outperforms all other approaches on the three domain splits, especially with respect to the EM metric, which we explain by the salient masking technique. Although $\text{BRENT}_{\text{reader}}$ was only trained on Wikipedia, the improvement in performance is significant also for questions in the news category. Naturally, $\text{NorBERT2}_{\text{50k}}$ also improves a bit compared to NorBERT2 on the Wikipedia split, but not by the same margin, and not at all on the news category. This indicates that $\text{BRENT}_{\text{reader}}$ actually learns to use context better, that this generalizes beyond

[9]These resources were made available to us through the EuroHPC JU project: https://www.lumi-supercomputer.eu/

Model	BSA F1 %	TSA F1 %
NorBERT2	$\mathbf{85.52}^{\pm 0.74}$	$\mathbf{47.58}^{\pm 0.49}$
$\text{NorBERT2}_{\text{50k}}$	$84.62^{\pm 0.50}$	$46.70^{\pm 0.65}$
$\text{BRENT}_{\text{reader}}$	$83.33^{\pm 0.47}$	$46.48^{\pm 0.26}$
$\text{BRENT}_{\text{Wiki}}$	$84.21^{\pm 0.37}$	$44.06^{\pm 0.73}$
$\text{BRENT}_{\text{Wiki;nri}}$	$84.18^{\pm 0.53}$	$43.38^{\pm 1.45}$
$\text{BRENT}_{\text{NoReC}}$	$84.35^{\pm 0.41}$	$43.55^{\pm 0.26}$
$\text{BRENT}_{\text{NoReC;nri}}$	$83.90^{\pm 0.56}$	$44.22^{\pm 0.42}$

Table 3: Results of the binary sentiment analysis task (BSA) on the NoReC_{sent} dataset and targeted sentiment analysis (TSA) on the NoReC_{fine} dataset. Evaluation is on the test set and is based on the best model found during training. Results are reported as the mean and standard deviation over five random seeds. nri stands for no re-indexing. The NoReC subscript represents the training dataset being used as the retrieval corpus.

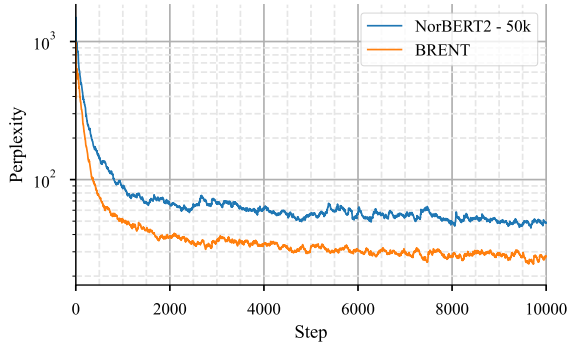


Figure 2: Training perplexity during the first 10 000 steps. The values are smoothed with an exponential moving average, using $\alpha = 0.99$.

the style of Wikipedia, and that this could not be achieved by simply training the same underlying LM for 50k additional steps on the same corpus with the same MLM setup.

4.4.2 Sentiment analysis

As for sentiment analysis, Table 3 shows that $\text{BRENT}_{\text{reader}}$ performs worse compared to the baseline of NorBERT2 on the binary sequence classification task, indicating that the continued pre-training with retrieval does not actually help for this task, but rather impedes performance. This is also the case for the $\text{NorBERT2}_{\text{50k}}$ model, albeit with a smaller impediment to performance, suggesting that it might be the continued training on Wikipedia reducing the performance of the models on this task. When retrieval is used, as can be seen in the bottom half of Table 3, the performance is better, but still short of the baseline. For TSA, the reader performs on par with the baselines but turning retrieval on substantially decreases performance.

When retrieving from a corpus during fine-tuning, our model retrieves reviews that are related with respect to inner product similarity, not necessarily sentiment. If classifying a negative review of a TV, our model could end up retrieving another review about some other electronic apparatus — which might be positive. This is clearly not helpful for the task at hand. In order to teach the retrievers to retrieve based on sentiment, we would need a bigger dataset to fine-tune on. Despite this, manual inspection shows that the retrieved contexts are sometimes very relevant for the query when the retrieval corpus is NoReC. When the retrieval corpus is Wikipedia, however, the contexts are of low relevance. Examples of queries and retrieved contexts for $\text{BRENT}_{\text{Wiki}}$ and $\text{BRENT}_{\text{NoReC}}$ on binary sentiment analysis (BSA) can be found in Appendix A.1.1 and Appendix A.1.2.

We also note that not re-computing the search index decreases performance. However, as performance is relatively similar, it might not be worth it as re-indexing is a lot more resource-demanding. With Wikipedia as the retrieval corpus on our computing setup, TSA fine-tuning takes about 7 hours with re-indexing, compared to 2.5 hours without.

4.4.3 Morpho-syntactic

Table 2 shows the results of the reader compared to the baselines on a series of Norwegian token-level labeling tasks. $\text{BRENT}_{\text{reader}}$ performs on par with the baseline models, which strengthens our hypothesis that the continued pre-training with retrieval does not impede the model's ability to perform morpho-syntactic tasks while simultaneously increasing performance on extractive QA. This claim is further supported by the fact that the same happens with $\text{NorBERT2}_{\text{50k}}$, which indicates that adding the retrieval is no worse than just continuing to do MLM over the same corpus for additional steps.

4.5 Analysis of the pre-training

Figure 2 shows the perplexity values of BRENT and $\text{NorBERT2}_{\text{50k}}$ during the first 10k steps of continued pre-training on the Wikipedia corpus. After the initial convergence phase, $\text{NorBERT2}_{\text{50k}}$ establishes itself on values around 40, while BRENT sits at around 20. As we do mainly salient masking, perplexity is a proxy for how well the models predict the correct named entities and dates. The difference between the two shows how retrieval is helpful for predicting masked entities.

5 Ablations

As with other retrieval-augmented LMs, BRENT is a pipeline model — consisting of multiple parts that interact according to a series of design choices that impact the outcome. Due to the computational cost of pre-training, it is not feasible to quantitatively determine the effect of all these choices, resulting in a poor understanding of some aspects of these models. To mitigate this, we study the effect of some of these choices with respect to the overall loss during pre-training with a series of ablations. We do this for a reduced number of steps, but with

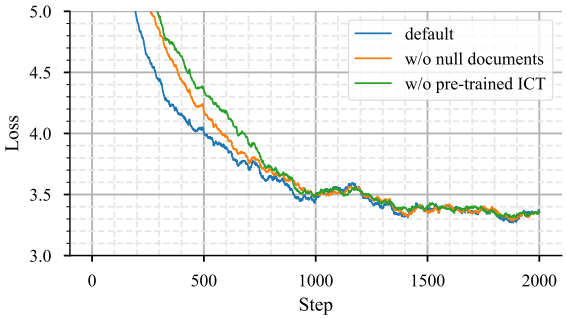


Figure 3: Loss curves of having no null document or no ICT warmup compared to the tested model. The values of all three runs are smoothed with an exponential moving average, using $\alpha = 0.99$.

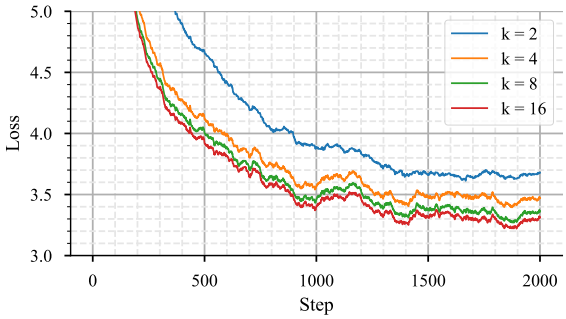


Figure 4: Loss curves of the first 2 000 training steps with varying number of retrieved documents k. The values are smoothed with an exponential moving average, using $\alpha = 0.99$.

the same retrieval corpus and with the same GPU setup as described in Section 3.3 and Section 4.3.

5.1 ICT

Figure 3 shows the effect of the ICT warmup task with respect to the loss for 2000 steps. When ICT is turned off, Enc_{doc} and $\text{Enc}_{\text{query}}$ are initialized with the same weights as the reader. As can be seen from the figure, the loss converges slower when the ICT task is not used, but it is quickly matching the setting when it is used. Guu et al. (2020) claims that without ICT one would encounter a cold-start problem where the retrieved documents will likely be unrelated to the query at the beginning of training, causing a cycle where the encoders do not receive meaningful gradients. We find that this is not the case and that the effect of ICT warmup is minimal.

5.2 The effect of the null document

As mentioned in Section 3.2, we use a parameter tensor initialized with all zeros for representing the null document. This is optimized jointly with the rest of the weights. Figure 3 shows how the model behaves when the null document is removed, which is done by making the probability of the null document zero, as compared to the test model which has it included. Contrary to Guu et al. (2020), we find that the effect of the null document is questionable. It makes sense to have a ”sink” to use when no retrieval is necessary, but we do not find the null document to fulfill this need.

5.3 Varying the number of documents to retrieve

Figure 4 shows how the number of retrieved documents influences training with respect to loss. For the first 2 000 training steps, $k = 16$ converges a bit quicker than the $k = 8$. However, we see that the result is minimal after that point, which is also the conclusion in Guu et al. (2020). Given that it is more computationally expensive to train with a higher k and that the gain of going from 8 to 16 is negligible, we keep k at 8.

6 Conclusion

We develop the first Norwegian retrieval augmented language model, BRENT, based on the REALM method proposed by Guu et al. (2020). The model uses an encoder-reader architecture, and we train it on a relatively small corpus consisting of 735k Wikipedia documents. In addition to the model itself, our contribution has been to demonstrate how the use of continued pre-training with retrieval benefits the context utilization of the reader, which we extract from the pipeline. The reader performs better than comparable baselines on the extractive QA task without losing performance on morpho-syntactic tasks. We also evaluate our full retriever model on sentiment analysis with two different corpora as the retrieval corpus, but here we observe a decrease in performance overall. Contrary to some previous work, our ablation studies find that the effect of having a null document and using ICT as a warmup task is minimal.

7 Future work

A future direction for our work is to study in greater detail how retrieval influences the language modeling task. In particular, we would like to train a retrieval model from scratch. Another direction, which has also been pointed out in related work, is to experiment with cross-lingual retrieval, especially in the case where the retrieval corpus is from a high-resource language. This would be useful in scenarios where a large knowledge source like English Wikipedia could be used to augment a lower resource language, like Norwegian, which does not have such an extensive source available.

Acknowledgements

Parts of the work documented in this publication have been carried out within the NorwAI Centre for Research-based Innovation, funded by the Research Council of Norway (RCN), with grant number 309834.

References

Uri Alon, Frank Xu, Junxian He, Sudipta Sengupta, Dan Roth, and Graham Neubig. 2022. Neuro-symbolic language modeling with automaton-augmented retrieval. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 468–485. PMLR.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Efficient nearest neighbor language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5703–5714, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sardana Ivanova, Fredrik Aas Andreassen, Matias Jentoft, Sondre Wold, and Lilja Øvrelid. 2023. NorQuAD: Norwegian Question Answering Dataset. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, Torshavn, Faroe Islands.

Zhengbao Jiang, Luyu Gao, Jun Araki, Haibo Ding, Zhiruo Wang, Jamie Callan, and Graham Neubig. 2022. Retrieval as attention: End-to-end learning of retrieval and reading within a single transformer. *arXiv preprint arXiv:2212.02027*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Fredrik Jørgensen, Tobias Aasmoe, Anne-Stine Ruud Husevåg, Lilja Øvrelid, and Erik Velldal. 2020. NorNE: Annotating Named Entities for Norwegian. In *Proceedings of the 12th Edition of the Language Resources and Evaluation Conference*, Marseille, France, 2020.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided supervision for OpenQA with ColBERT. *Transactions of the Association for Computational Linguistics*, 9:929–944.

Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-scale contextualised language modelling for Norwegian. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 30–40, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob

Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Lilja Øvrelid and Petter Hohle. 2016. Universal Dependencies for Norwegian. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, pages 1579–1585, Portorož, Slovenia.

Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2020. A fine-grained sentiment dataset for Norwegian. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5025–5033, Marseille, France. European Language Resources Association.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6648–6662, Online. Association for Computational Linguistics.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

Per Erik Solberg, Arne Skjærholt, Lilja Øvrelid, Kristin Hagen, and Janne Bondi Johannessen. 2014. The Norwegian Dependency Treebank. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, Reykjavik, Iceland.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Erik Velldal, Lilja Øvrelid, Eivind Alexander Bergem, Cathrine Stadsnes, Samia Touileb, and Fredrik Jørgensen. 2018. NoReC: The Norwegian review corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

A Appendix

A.1 BSA retrieval examples

A.1.1 Wikipedia examples

Examples of retrieved contexts from the $\text{BRENT}_{\text{Wiki}}$ model fine-tuned on the BSA task with Wikipedia as the retrieval corpus.

- Query: *Men så kommer de skjærende lydene* 'But then the squeaky sounds appear'
  - Retrieved context: *Øynene* 'The eyes'.
- Query: *Broen går seg vill i sitt eget ønske om å vært artsy* '"The Bridge" is lost in its own wish to be "artsy"'
  - Retreived context: *Sverige* 'Sweden'

A.1.2 NoReC examples

Examples of retrieved contexts from the $\text{BRENT}_{\text{NoReC}}$ model fine-tuned on the BSA task with the $NoReC$ dataset as the retrieval corpus.

- Query: *Men så kommer de skjærende lydene* 'But then the squeaky sounds appear'
  - Retrieved context: *Allerede under første låt får vi slengt alle klisjeene i trynet.* 'Already during the first song we are hit in the face with all the clichés'.
- Query: *Begeistringen var uvanlig stor og applausen deretter da det 70 minutters lange verket var fullført* 'The enthusiasm was unusually great and so was the applause that followed when the 70-minute long piece was over'.
  - Retreived context: *En helt utrolig konsertopplevelse* 'A wonderful concert experience'.
- Query: *Å være eksperimentell er ikke positivt i seg selv; de mange sjangrene og retningene i musikken gjør helehetsinntrykket rotete og meningsløst* 'Being experimental is not positive in and of itself; the many genres and directions makes the music seem messy and meaningless'.
  - Retrieved context: *Automatisk to-soners klimaanlegg* 'Automatic two-zone aircondition'.

A.2 Model

The hyperparameters used for the continued pretraining can be found in Table 4.

A.3 Hyperparameters

A.3.1 NorQuAD

For comparison, we use the same set of hyperparameters as in Ivanova et al. (2023), visible in Table 5.

A.3.2 Sequence labeling

For the task of targeted sentiment analysis, we fine-tune and report the average test results over five runs, from the epoch providing the best results on the development set. Hyperparameters can be found in Table 6.

A.4 Binary Sentiment Analysis

For the task of binary sentiment analysis, we fine-tune for three epochs and select the best model based on the development set's f1 score. We average our test results over five runs. Hyperparameters can be found in table Table 7.

A.5 Morpho-syntactic

For morpho-syntactic tasks, we fine-tune for 10 epochs and select the best model based on the average performance on the development split. We average our test results over five runs. Hyperparameters can be found in Table 8.

Hyperparameter	Value
Number of parameters	125M
Number of attention heads	12
Number of layers	12
Hidden dimension (h_{dim})	768
Activation function	GeLU
Vocabulary size	50104
Seed	42
Dropout	0.1
lr	$2 * 10^{-5}$
Weight decay	0.1
Batch size	1024
k	8
re-indexing frequency	100 steps
Steps	50k
Scheduler	Cosine with warmup
Warmup	800 steps
Final lr	$2 * 10^{-6}$
Optimizer	AdamW
Index	FlatIndexIP

Table 4: Hyperparameters for the continued pre-training of both BRENT and NorBERT2. k represents the number of documents retrieved including the null document.

Hyperparameter	Value
Batch size	16
Epochs	3
lr	$5 * 10^{-5}$
Scheduler	Linear
Optimizer	AdamW
Seeds	$[42, 437, 4088, 3092, 9720]$

Table 5: Hyperparemeters for fine-tuning on the NorQuAD dataset

Hyperparameter	Value
Batch size	32
Epochs	8
$\text{lr}_{\text{reader}}$	$5 * 10^{-5}$
$\text{lr}_{\text{retriever}}$	$1.5 * 10^{-4}$
k	4
re-indexing frequency	100 steps + end of epoch
Scheduler	Linear
Optimizer	AdamW
Seeds	$[101, 202, 303, 404, 505]$

Table 6: Hyperparemeters for fine-tuning on the NoREC$_{fine}$ dataset. k represents the number of documents retrieved including the null document. The $\text{lr}_{\text{reader}}$ is for both the retrieval and non-retrieval models.

Hyperparameter	Value
Batch size	32
Epochs	3
$\text{lr}_{\text{reader}}$	$1 * 10^{-5}$
$\text{lr}_{\text{retriever}}$	$3 * 10^{-5}$
k	4
re-indexing frequency	100 steps + end of epoch
Scheduler	Cosine
Optimizer	AdamW
Seeds	$[42, 456, 78463, 27485, 34586]$

Table 7: Hyperparemeters for fine-tuning on the NoREC$_{sent}$ dataset. k represents the number of documents retrieved including the null document. The $\text{lr}_{\text{reader}}$ is for both the retrieval and non-retrieval models.

Hyperparameter	Value
Batch size	32
Epochs	10
$\text{LR}_{\text{reader}}$	$1 * 10^{-4}$
LR_{heads}	$1 * 10^{-3}$
Scheduler	Cosine
Optimizer	AdamW
Seeds	$[1234, 2345, 3456, 4567, 5678]$

Table 8: Hyperparemeters for fine-tuning on the morpho-syntactic tasks. k represents the number of documents retrieved including the null document. The learning rate is different for the fine-tuned language model and for the classification heads.

Machine vs. Human: Exploring Syntax and Lexicon in German Translations, with a Spotlight on Anglicisms

Anastassia Shaitarova, Anne Göhring, Martin Volk
Department of Computational Linguistics, University of Zurich
{*shaita, goehring, volk*}*@cl.uzh.ch*

Abstract

Machine Translation (MT) has become an integral part of daily life for millions of people, with its output being so fluent that users often cannot distinguish it from human translation. However, these fluid texts often harbor algorithmic traces, from limited lexical choices to societal misrepresentations. This raises concerns about the possible effects of MT on natural language and human communication and calls for regular evaluations of machine-generated translations for different languages. Our paper explores the output of three widely used engines (Google, DeepL, Microsoft Azure) and one smaller commercial system. We translate the English and French source texts of seven diverse parallel corpora into German and compare MT-produced texts to human references in terms of lexical, syntactic, and morphological features. Additionally, we investigate how MT leverages lexical borrowings and analyse the distribution of anglicisms across the German translations.

1 Introduction

Advanced text generation tools such as ChatGPT[1] and Machine Translation (MT) are used by millions of people every day. With the scope of human exposure to machine-generated texts ever-growing, these tools possess the potential to have an impact on natural language. The scientific community is yet to establish a research paradigm suitable for the assessment of this impact. In the meantime, we investigate generated texts and compare them to human-produced texts. In the present paper, we focus on machine translation for the German language.

Translation study scholars long established that any translation has the potential to affect the target language (TL). First, Gellerstam (1986) noticed that the translation process leaves "fingerprints" in the TL translation and named the resulting "fingerprinted" language *translationese*. The common characteristics of (human) translated text became formalized as *translation universals* or even *translation laws* (Toury, 1995; Baker, 1995). These patterns include simplification, explicitation, overall normalization, and standardization. Moreover, the source text often "shines through" (Teich, 2003) in the target text. Kranich (2014) hypothesised that these patterns persevere beyond any given translation, reappearing in texts later produced by the native TL writers. In fact, Kranich conceptualized translation as a virtual place where languages come into contact and change as a result. The severity of change is defined by many factors, including the intensity and length of exposure.

Human exposure to MT output is expected to increase, and the global MT market is steadily growing[2]. Machine-translated texts are used in almost all spheres of life, from schools (Morton, 2022), to academic publishing (Anderson, 2021), to governments (Jaun, 2019; Dalzell, 2020; Percival, 2022), and even hospitals and courts (Nunes Vieira et al., 2020; Khoong and Rodriguez, 2022; Kapoor et al., 2022). New MT engines continue to enter the market and language coverage has reached over 200 languages (Siddhant et al., 2022) and tens of thousands language pairs across all MT systems[3].

Several researchers already started to investigate the sociolinguistic impact of machine translation. For instance, MT use has been shown to have a direct and long-lasting effect on the syntactic production of language learners (Resende and Way, 2021). While producing highly fluent

[1] openai.com/blog/chatgpt

[2] statista.com/statistics/748358/worldwide-machine-translation-market-size

[3] State of Machine Translation 2021 report

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 215–227

translations, the MT output can suffer from simplification and even impoverishment (Vanmassenhove et al., 2021; Vanroy, 2021). Moreover, MT models are known to overgeneralize and amplify societal biases (Prates et al., 2020; Farkas and Németh, 2022; Troles and Schmid, 2021; Vanmassenhove et al., 2021; Hovy et al., 2020). When it comes to the analysis of commercial MT systems, however, most research focuses on the English output of Google Translate[4] with rare mentions of other translation engines (Almahasees, 2018; Aiken, 2019; Matusov, 2019; Webster et al., 2020; Hovy et al., 2020; Brglez and Vintar, 2022).

In our paper, we explore the output of three widely used engines (Google, DeepL, Microsoft Azure) and one smaller commercial system. We work with translations from English and French to German, a morphologically and syntactically complex language. We use seven different corpora (Section 2) and a battery of evaluation metrics which examine the texts on lexical, syntactic, and morphological levels (Section 3). Moreover, in Section 3.3, we scrutinize the translations from a novel angle, by looking at the distribution of anglicisms in the German texts - the process of lexical borrowing being a crucial feature of language change and evolution (Miller et al., 2020).

2 Data

2.1 Selection of test corpora

We follow three criteria in the selection of our test corpora. First, we experiment with different domains. Second, we avoid back-translation and translationese, since they interfere with evaluation metrics and might skew the results (Toral et al., 2018; Zhang and Toral, 2019; Graham et al., 2020). However, it is difficult to find parallel corpora with a clearly-marked source language.

Finally, to prevent cross-contamination of train and test data, we work with test corpora that have not been used as training data by commercial MT systems. Since the MT companies do not disclose the composition of their training corpora, we follow a common-sense assumption that all large, publicly available parallel corpora with a dated online presence have been used for MT training. Following this logic, we refrained from using Europarl, ParaCrawl, and other similar multilingual datasets. Instead, we collected seven corpora that mostly comply with our prerequisites. We describe them in detail in the following subsections and give a general overview in Table 1.

2.1.1 WMT21 and WMT22

Our first logical choice of data was the test sets for the Conference on Machine Translation[5] (WMT), since they are used for the evaluation of MT systems, and therefore consciously kept out of training data. The test sets from 2021 and 2022 contain professional translations "from scratch", without back-transaltions or post-editing.

The WMT21 News Test Set[6] is a collection of online news from 2020 aligned with professional human translations (Akhbardeh et al., 2021). The original texts are collected online in English from various American, English, and Australian newspapers as well as from Al Jazeera English, allafrica.com (a news aggregation site), two Indian news sources, and euronews-en.com, a television news network headquartered in France.

The novelty of WMT22 (Kocmi et al., 2022) is that the data comes in equal parts from 4 different domains: news, e-commerce, conversation, and social media. The test set contains roughly 500 sentences for each domain. The quality of the test data is controlled manually to avoid noise and inappropriate content.

2.1.2 Tatoeba

Tatoeba[7] is a non-profit association which maintains an online open depository of crowd-sourced original and translated sentences in multiple languages. The downloadable set of sentences is updated every week. We selected 1777 most recent English-German pairs dating between September and December 2022. We picked only those pairs where the source English sentences are indicated as original text and translated into German by users claiming a native or high level of German.

2.1.3 transX

We obtained a parallel corpus of human English-German translations containing non-sensitive data from a private translation company. Despite some of the texts being featured in the company's blog, the translation memory has not been made available to the public. The corpus contains texts about translation, editing, general business, technology, and other related topics.

[4] translate.google.com

[5] www.statmt.org/wmt22/

[6] github.com/wmt-conference

[7] tatoeba.org

corpus	domain	src lang	sent pairs	one2one	tokens	src-tgt	remarks
WMT 21	news	en	1,002	814	27,937	web-prof	–
WMT22	mixed	en	2,037	1,850	39,164	web-prof	–
Tatoeba	mixed	en	1,777	1,685	16,285	crowd-crowd	trust-based
transX	mixed/tech	en	1,164	965	20,359	unk-prof	urls, jargon
Jane Eyre	classic lit	en	8,784	3,964	229,283	prof-prof	seen by MT
Text+Berg	alpine texts	fr	22,662	21,353	465,776	mixed-unk	OCR errors
CS Bulletin	mixed	en	59,348	54,840	1,164,694	prof-prof	back-translated?

Table 1: Overview of the corpora. Number of tokens is indicated for the original source sentences.

2.1.4 Jane Eyre

The novel Jane Eyre by Charlotte Brontë is part of the Gutenberg Project dataset. It was aligned with its German translation by András Farkas[8] and made available on OPUS. Classical literature provides certainty about the original source language, yet is counteracted by a high likelihood that it has been seen by the commercial English-German MT models during training. Published in 1847, Jane Eyre features some archaic language and spelling.

2.1.5 CS Bulletin

The Credit Suisse Bulletin corpus (Volk et al., 2016) is a digitized diachronic collection of texts from the world's oldest banking magazine, published by Credit Suisse[9]. The corpus contains parallel texts in German, French, Italian, and English, and covers topics pertaining to economy, culture, sport, entertainment, etc. We selected the German-English PDF subcorpus ranging from 1998 to 2017[10]. There is no proof of the source language, and we can only assume that German was the source of most articles since Credit Suisse originated in the German-speaking part of Switzerland. Therefore, the CS Bulletin corpus here mostly represents back-translated texts.

2.1.6 Text+Berg

Text+Berg is a diachronic corpus of Alpine texts predominantly written by Swiss mountaineers and spanning from 1864 to 2009[11] (Volk et al., 2010; Göhring and Volk, 2011). We included all French-German parallel articles published since 1957. Due to incomplete metadata, we limited our selection to articles that explicitly stated the source language as French in the German translation, such as "Aus dem Französischen von" (*[Translated] from French by*), while excluding French articles that were translated from a language other than French, such as "Traduit de l'anglais par" (*translated from English by*).

2.2 Preprocessing and Translation

We translated all source texts automatically into German using four commercial MT systems: Google Translate, DeepL, Microsoft Azure, and a small private commercial MT engine specializing in German (here: mtX). The translations were performed in November 2022. As a point of reference, we provide the translation quality scores produced by COMET (Rei et al., 2020) in Table 4. This metric draws information from both source and reference texts, and captures surface and semantic similarities. We provide more conventional SacreBLEU scores (which happen to show a similar pattern) in the Appendix A.

corpus	azure	deepl	google	mtX
WMT21	53.51	**57.77**	52.50	49.07
WMT22	62.06	**64.19**	62.24	58.58
Tatoeba	71.07	**74.13**	72.89	69.92
transX	59.69	**63.18**	59.09	56.82
JaneEyre	21.23	**29.57**	24.14	17.73
CSBull	68.30	**69.52**	68.94	66.78
Text+Berg	28.78	**41.32**	34.38	31.30

Table 2: COMET-DA_2020 scores per MT system on full-sized corpora. The best values are in **bold**.

Since both the Credit Suisse and Text+Berg corpora contain OCR errors and poor sentence alignments, we performed an additional alignment step. We identified the most probable sentence pairs using LASER margin-based sentence alignment (Artetxe and Schwenk, 2019) with a rather strict margin criterion value of 1.2. We tokenized all texts using the Spacy-UDPipe Tokenizer[12].

The tasks of syntactic comparison and automatic anglicism analysis require precise word

[8] farkastranslations.com/bilingual_books.php

[9] credit-suisse.com/cn/en/content-hub/bulletin.htm

[10] pub.cl.uzh.ch/projects/b4c

[11] textberg.ch

[12] github.com/TakeLab/spacy-udpipe

alignment, which is complicated in sentence pairs with a one-to-many translation. For these tasks, we created a subsection of each corpus with only one-to-one sentence alignments. Since sentence segmentation and the choice of one-to-one or one-to-many sentences differ across translations, we selected only those sentence pairs from each translation of a corpus, where the source language sentences are the same as the ones in the one-to-one human translation pairs. In other words, we made an intersection of all translation pairs (human and MT) with an anchor on the human translation. The WMT datasets contain several human references. Here, we base our filtering on the translation that exhibits the smallest number of n-to-n pairs: WMT21 - reference C and WMT22 - reference A. The number of sentences in these subcorpora can be found in Table 1.

3 Metrics and Findings

We used several metrics to analyze the available translations in terms of their lexical, syntactic, and morphological features.

3.1 Lexical analysis

Lexical diversity We investigated our texts with respect to lexical diversity using a variety of metrics within the BiasMT[13] tool developed by Vanmassenhove et al. (2021). We used the Type-Token Ratio (TTR) metric, which provides a general overview of lexical diversity in a text. Since TTR is known to skew results in long texts, we also employed the measure of textual lexical diversity (MTLD), which assesses the length of word sequences with a specific level of TTR (McCarthy, 2005), as well as Yule's K (Yule, 1944), which is resilient to text length fluctuations while reflecting the repetitiveness of the data.

Although the results of our investigation show higher diversity values in human translations, several MT systems produced competitively diverse translations for some of the corpora. The mtX system scored the highest TTR values on WMT21, WMT22, Jane Eyre, and transX. It scored the highest MTLD on WMT21, and WMT22. Google scored the highest Yule's I and MTLD on the Jane Eyre translation (full results in Appendix B).

Sophistication Another way to examine the lexical diversity of a text is to measure its sophistication. This involves measuring how much text is filled with the most and the least frequent words. A lexically diverse text usually has a lower percentage of tokens that belong to the $1,000$ most frequent words. Subsequently, there would be a larger percentage of rare and unusual words in such a text. In our experiments, the sophistication results show the same pattern as the lexical diversity metrics. Human translations prove to be most lexically diverse in all the corpora except WMT22 and Jane Eyre where mtX exhibits the highest diversity (full results in Appendix C).

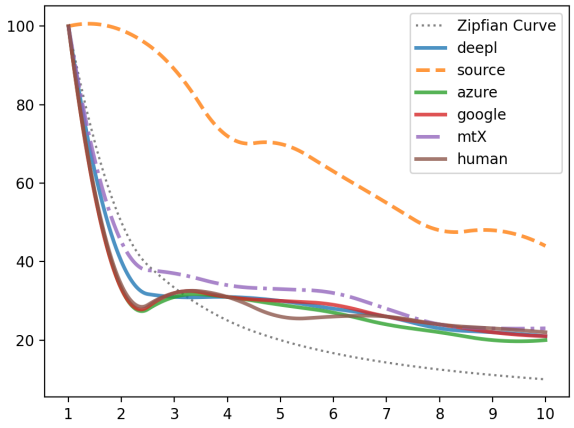


Figure 1: The Zipfian distribution of the English text and its translations in the Tatoeba corpus. The mtX output shows higher diversity of the medium frequency words than the other MT systems.

Inflectional paradigms Additionally, we assessed the morphological complexity and richness of each text using Shannon entropy and Simpson's diversity. Shannon entropy measures the surprisal level within each lemma's inflectional paradigm. For example, the distribution of the word forms for the German lemma *Problem* can be the following in Google's translation: {Problem:7, Probleme:3, Problemen:1, Problems:0}. If the word forms are distributed more evenly in the human translation ({Problem:4, Probleme:2, Problemen:2, Problems:3}), then the entropy for this lemma is higher than in the text translated by Google. The scores are averaged over all lemmas that appear at least as two different word forms in a corpus. Simpson's diversity reflects variability in categorical data. Higher scores indicate homogeneity, while lower scores denote diversity.

Vanmassenhove et al. (2021) observed that machine-translated English, French, and Spanish texts were less morphologically diverse than the texts used for training the same MT systems. We

[13] github.com/dimitarsh1/BiasMT

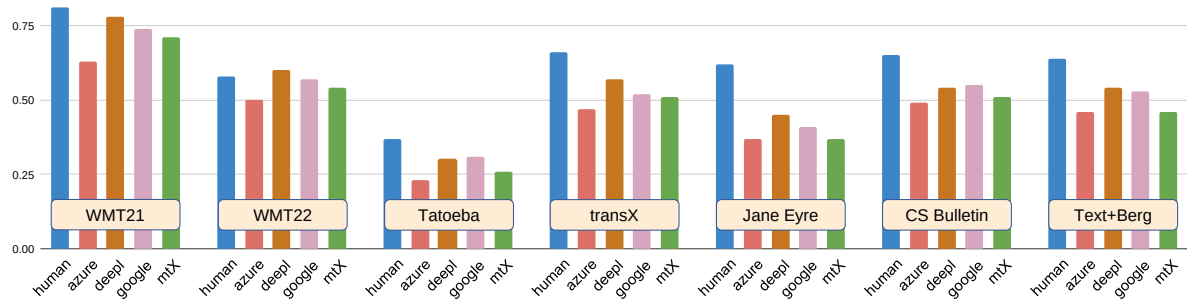


Figure 2: The measure of syntactic equivalence is calculated as the ratio of cross-alignments to the total number of word alignments. The higher score indicates more syntactically creative translation.

compare human and machine-translated texts and notice that commercial MT systems produce German texts that are comparable to human translations in terms of morphological richness. The mtX system scored higher values for the Tatoeba and the CS Bulletin corpora. DeepL produced the most diverse inflectional distributions in the translations of Jane Eyre and Text+Berg. Microsoft Azure exhibited the richest morphology in the transX corpus (see Appendix D).

In summary, our results show that the human translation and the MT output of the German-specialized company exhibit the highest scores for lexical diversity and sophistication. Our morphological richness results differ from the standard lexical diversity scores with more than one MT system exhibiting higher scores than the human translations.

This trend fluctuates slightly across the domains since each corpus has its own unique features. Text+Berg and CS Bulletin are large, diverse corpora with multiple writers, translators, OCR errors and specialized terminology. Tatoeba's sentences are crowd-sourced and the translators are encouraged to provide multiple translation variants. Assuming that MT tends to standardize, the lower MT diversity scores are not surprising in these corpora, although the morphological results show a different picture. The Jane Eyre and transX corpora are homogeneous in terms of domain and terminology. Here, some MT systems score higher than human texts in terms of all types of diversity.

Figure 1 illustrates lexical differences in the translations of the Tatoeba corpus using Zipf's rank-frequency distribution law. Duplicate sentences were left in for both languages. The graph demonstrates how the output of the German-specialized MT system exhibits higher diversity for mid-range frequency words, while all the translations are less diverse than the original text. Based on our results, we may infer that lexical impoverishment will not be the main issue with the machine-translated texts in the future. MT is improving rapidly for many languages, having access to more training data, and employing new decoding methods which control the diversity of the output. The quality and adequacy of translation notwithstanding, specialized systems can be tuned to produce lexically and morphologically rich texts.

3.2 Syntactic equivalence

We used the ASTrED tool[14] (Vanroy, 2021; Vanroy et al., 2021) to analyze the syntactic differences between texts. By dividing the number of cross-aligned words by the total number of word alignments, we obtained a measure of syntactic equivalence between the source text and its translations. The side-by-side results for all the corpora in Figure 2 clearly demonstrate that human translators exhibit greater syntactic creativity compared to any of the MT systems. These findings align with the results published by researchers for other language pairs (Tezcan et al., 2019; Webster et al., 2020; Vanroy, 2021).

Out of all our commercial MT systems, DeepL syntactically diversifies the output the most, while the other systems rather mimic the syntax of the source sentence, like in this example from the WMT21 corpus:

> **Eng**: Couple MACED at California dog park
> **Human**: Angriff mit Pfefferspray auf ein Paar in einem Hundepark in Kalifornien
> **DeepL**: Ehepaar wird in kalifornischem Hundepark angegriffen
> **Other MTs**: Paar MACED im kalifornischen Hundepark

[14] github.com/BramVanroy/astred

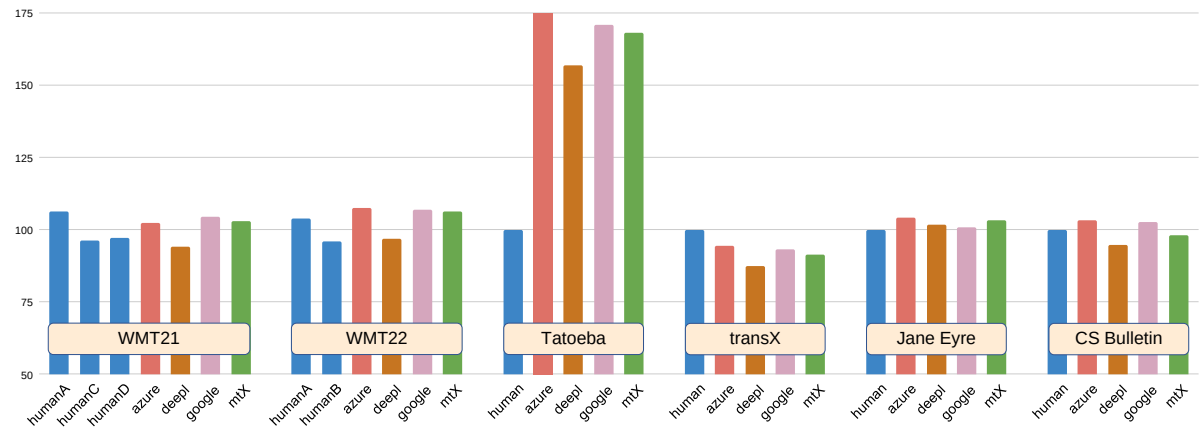


Figure 3: Distribution of anglicisms in different translations across corpora. The number of anglicisms in the human translations is taken as 100%.

Appendix E shows the translations of all 20 MT systems from the competition along with those of Google, Azure, and mtX. All of them mirror the syntax of the source sentence, whereas human translators and, to a certain extent, DeepL take liberty with the sentence structure.

3.3 Exploration of anglicisms

Lexical borrowings, the transfer of words from one language to another, is a productive mechanism of word formation and a catalyst of language evolution. Borrowings emerge from language contact, a universal linguistic phenomenon. They appear in all languages and can constitute a high percentage of lexical items. Identification of borrowings is important in lexicography, comparative linguistics, and some NLP downstream tasks, yet there is no reliable way to identify them automatically (Miller et al., 2020; List and Forkel, 2021).

We focus on English borrowings in German, known as anglicisms. The number of anglicisms in German is continuously growing. Reportedly, every 600th word in German could be identified as an anglicism in 1954. In 1964, it became every 200th word; in 1994, every 145th; and in 2004, every 85th (Engels, 1976; Burmasova, 2010). There is a notable societal push against this process or at least concerns about the future of the German language[15]. The investigation of this phenomenon can provide valuable insights into the role of MT in language development. We assess the extent to which MT language models participate in the anglicization of German. To the best of our knowledge, this is the first investigation of this kind.

There are many different ways to classify anglicisms in German: by topic, by type of surface form assimilation ("most anglicisms introduced since 1945 retain their English orthography" (Coats, 2019, p.273)), by level of assimilation (Eindeutschung), etc. Often anglicisms are classified into words indicating either new concepts (ergänzende Anglizismen, Bedürfnislehnwörter) or existing concepts (differenzierende (or verdrängende) Anglizismen[16], Luxuslehnwörter (Carstensen, 1965)). Since anglicisms continuously pour into the language but do not always stay, we work with the items that have mostly settled in German. We collected $4,832$ established anglicisms from a dedicated Wikipedia page[17], disregarding "false friends".

To avoid false positives, we filtered out certain homonyms, such as "Tag" (*day*) and "Gang" (*passageway*), and removed the word "in" which occurs in the lexicalized phrase "in sein" (*to be in*). Additionally, we excluded some corpus-specific anglicisms, for example "Credit" in the Credit Suisse Bulletin corpus, or "Miss" in the Jane Eyre corpus. The human translation of Jane Eyre contains an old, pre-1996 spelling of "Miss" as "Miß", which is not on the list of anglicisms.

We customized our search to catch different spelling variations of certain anglicisms (for example: *fairtrade, fair-trade, fair trade*). We to-

[15] Mind your language: German linguists oppose influx of English words; Denglisch – Deutsch oder Englisch?

[16] contify.de/glossar/richtig-schreiben/was-sind-anglizismen

[17] de.wiktionary.org: last update 12.06.2019; scraped in April, 2022

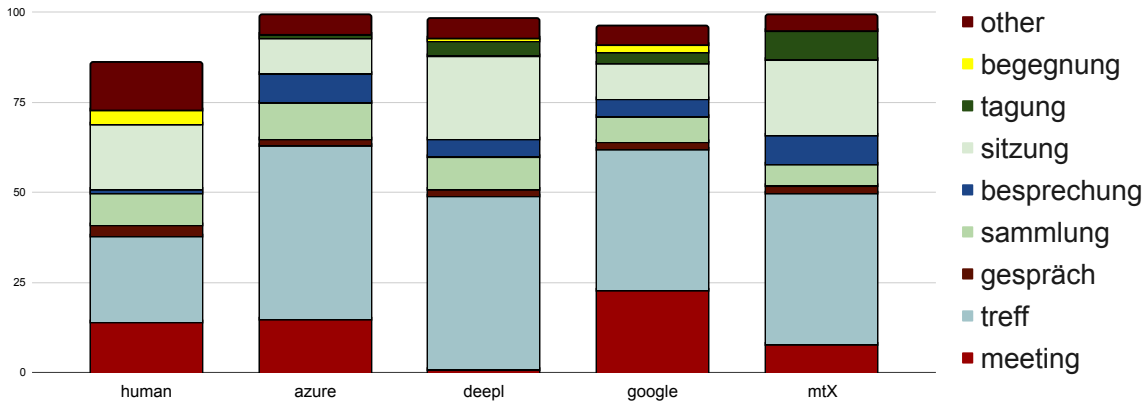


Figure 4: Distribution of lemmas for the translation variants of the anglicism "meeting" in the CS Bulletin corpus. The lemma "meeting" appears in the English text 119 times. The missing occurrences can be attributed to poor alignments.

kenized the texts with the Spacy UDpipe tool and matched anglicisms from our list to tokens, lemmas, and multiword units. Additionally, we looked for anglicisms inside German compound words. We used the Compound Split tool[18] to separate the components, and matched each component against the list of anglicisms.

We employed language detection on the produced word components to compensate for insufficient or inadequate splitting. However, language detection is not a reliable method for the identification of anglicisms. Thus, we collected the resulting alleged non-anglicisms from all the corpora into one list and manually filtered out true anglicisms. The example below shows words that were correctly and incorrectly identified as false positives of the anglicism *fan*:

true: fangen, fandest, Stefan, Fannie
false: Fanbasis, Autofan, Fanbild

The final list contained 342 entries, including words like *musstest* and *könntest* (falsely detected anglicism *test*); *gängig* (gig), *dadurch* (dad), *Psychologin* (gin), *hitzig* and *Hitler* (hit), etc.

Figure 3 shows the full distribution of anglicisms in all the translation versions across all corpora. The number of anglicisms in the human translations is taken as 100%. All other distributions are shown as relative to the human translation. Since the WMT corpora have several human references, the average of their scores is taken as a hundred percent mark.

[18] pypi.org/compound-split/

While we consider the human usage of anglicisms to be the gold standard, the distributions predictably vary even among translators. Similarly, this variability occurs among the MT systems as well. Some trends are noticeable, however. For example, DeepL produces fewer anglicisms than the three other systems, while Microsoft Azure tends to anglicize its output. Figure 4 provides a distribution of translation variant lemmas for a frequent anglicism *meeting* in the CS Bulletin corpus. It shows how this anglicism barely appears in the DeepL output. Nevertheless, the overall distribution of translation variants appears to be more even in the human translation, whereas the MT systems lean towards one particular lemma (here: *treff*).

While most corpora show gentle fluctuations in the anglicism distribution across the systems, we observe a striking difference between the human and machine translations for Tatoeba. This might be due to the fact that all translations are provided by crowd-sourced volunteers, who are eager to show their love and knowledge of German. The distribution of anglicisms in this corpus has a long tail of anglicisms that were avoided by the human translators, but employed by MT: *job, meeting, online, team, internet, baby, flirt, teenager,* etc.

Conversely, the human translations of a small translation company (the transX corpus) exhibit consistently more anglicisms than the output of all other MT systems. This might have to do with the fact that professional translators follow a consistency protocol appropriate to the client's business domain (here: tech). MT systems, on the other hand, maintain a steady degree of diversification.

4 Conclusion

This paper provides a corpus linguistic analysis of different translations, performed by humans and machines, in seven corpora from different domains. We looked at the texts mostly on a micro-level, measuring their lexical and syntactic properties, such as type-token ratio, morphological richness, and syntactic versatility. Additionally, we examined the distribution of translation variants for English lexical items that have entered the German language as borrowings or loan words.

Previous research emphasized that machine-produced texts suffer from standardization, simplification, and monotonicity. On one hand, our results confirm these findings in terms of syntax (section 3.2). On the other hand, we show that machine translation is becoming less of a culprit when it comes to lexical impoverishment of language. Some commercial MT systems are capable of generating German texts with levels of lexical and morphological richness similar to those produced by human translators (Section 3.1). Of course, these results reflect only one aspect of translation quality, and our automatic scores - as imperfect as they are - suggest that DeepL, not mtX, is the most reliable system for German translations (see Table 4).

Finally, we note that the standard lexical and syntactic metrics might be getting less informative for the linguistic assessment of MT as the technology continues to improve. Alternatively, automatic evaluation of lexical borrowings, such as anglicisms in German, can provide a good opportunity to assess the appropriateness of MT use. The distribution of borrowings is directly related to the quality and purpose of translation. Our results indicate that certain machine translation systems tend to produce fewer anglicisms compared to other systems (Section 3.3). In general, human translators adjust the use of anglicisms according to the domain, while the MT systems produce mostly consistent, system-specific distributions.

As machine translation improves and becomes more widespread, it will likely play a role in the (de-)anglicization of German. To mitigate this impact on German, more research is needed to accurately identify linguistic borrowings. Overall, our study sheds light on the current state of machine translation, laying the groundwork for investigating the potential impact that generated texts might have on human language.

Acknowledgments

This research was funded by the National Centre of Competence in Research "Evolving Language", Swiss National Science Foundation (SNSF) Agreement 51NF40_180888

References

Milam Aiken. 2019. An Updated Evaluation of Google Translate Accuracy. *Studies in Linguistics and Literature*, 3:p253.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 Conference on Machine Translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Zakaryia Mustafa Almahasees. 2018. Assessment of Google and Microsoft Bing Translation of Journalistic Texts. *International Journal of Languages, Literature and Linguistics*, 4(3):231–235.

Porter Anderson. 2021. During Frankfurt: Springer Nature Offers Auto-Translation for Research. *Publishing Perspectives*.

Mikel Artetxe and Holger Schwenk. 2019. Margin-based Parallel Corpus Mining with Multilingual Sentence Embeddings. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203. ArXiv: 1811.01136.

Mona Baker. 1995. Corpora in Translation Studies: An Overview and Some Suggestions for Future Research. *Target. International Journal of Translation Studies*, 7(2):223–243. Publisher: John Benjamins.

Mojca Brglez and Špela Vintar. 2022. Lexical Diversity in Statistical and Neural Machine Translation. *Information*, 13(2):93. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

Svetlana Burmasova. 2010. *Empirische Untersuchung der Anglizismen im Deutschen am Material der Zeitung Die WELT (Jahrgänge 1994 und 2004)*. Bamberg University Press. Accepted: 2011-01-31 00:00:00.

Broder Carstensen. 1965. *Englische Einflüsse auf die deutsche Sprache nach 1945*. Number ARRAY(0x56395ff24608) in Jahrbuch für Amerikastudien. Winter, Heidelberg.

Steven Coats. 2019. Lexicon geupdated: New German anglicisms in a social media corpus. *European Journal of Applied Linguistics*, 7(2):255–280. Publisher: De Gruyter Mouton Section: European Journal of Applied Linguistics.

Stephanie Dalzell. 2020. Google Translate used over professional translators for Government's official COVID-19 messaging. *ABC News*.

Barbara Engels. 1976. *Gebrauchsanstieg der lexikalischen und semantischen Amerikanismen in zwei Jahrgängen der "Welt" (1954 und 1964): e. vergl. computerlinguist. Studie zur quantitativen Entwicklung amerikan. Einflusses auf d. dt. Zeitungssprache*. Peter Lang. Google-Books-ID: EKtbAAAAMAAJ.

Anna Farkas and Renáta Németh. 2022. How to measure gender bias in machine translation: Real-world oriented machine translators, multiple reference points. *Social Sciences & Humanities Open*, 5(1):100239.

Martin Gellerstam. 1986. Translationese in Swedish novels translated from English. In *undefined*.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2020. Statistical Power and Translationese in Machine Translation Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 72–81, Online. Association for Computational Linguistics.

Anne Göhring and Martin Volk. 2011. The Text+Berg Corpus An Alpine French-German Parallel Resource. pages 97–102, Montpellier, France. ATALA.

Dirk Hovy, Federico Bianchi, and Tommaso Fornaciari. 2020. "You Sound Just Like Your Father" Commercial Machine Translation Systems Include Stylistic Biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1686–1690, Online. Association for Computational Linguistics.

René Jaun. 2019. Bundesverwaltung erwirbt DeepL-Lizenzen. *Netzwoche*.

Ravish Kapoor, German Corrales, Manuel P. Flores, Lei Feng, and Juan P. Cata. 2022. Use of Neural Machine Translation Software for Patients With Limited English Proficiency to Assess Postoperative Pain and Nausea. *JAMA Network Open*, 5(3):e221485.

Elaine C. Khoong and Jorge A. Rodriguez. 2022. A Research Agenda for Using Machine Translation in Clinical Medicine. *Journal of General Internal Medicine*, 37(5):1275–1277.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 Conference on Machine Translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Svenja Kranich. 2014. Translations as a Locus of Language Contact. In Juliane House, editor, *Translation: A Multidisciplinary Approach*, Palgrave Advances in Language and Linguistics, pages 96–115. Palgrave Macmillan UK, London.

Johann-Mattis List and Robert Forkel. 2021. Automated identification of borrowings in multilingual wordlists. *Open Research Europe*, 1:79.

Evgeny Matusov. 2019. The Challenges of Using Neural Machine Translation for Literature. In *Proceedings of the Qualities of Literary Machine Translation*, pages 10–19, Dublin, Ireland. European Association for Machine Translation.

Philip M McCarthy. 2005. *An assessment of the range and usefulness of lexical diversity measures and the potential of the measure of textual, lexical diversity (MTLD) - ProQuest*. Ph.D. thesis, University of Memphis, Memphis, Tennessee, USA.

John E. Miller, Tiago Tresoldi, Roberto Zariquiey, César A. Beltrán Castañón, Natalia Morozova, and Johann-Mattis List. 2020. Using lexical language models to detect borrowings in monolingual wordlists. *PLOS ONE*, 15(12):e0242709. Publisher: Public Library of Science.

Neal Morton. 2022. Translating a quarter of a million text messages for families. *The Hechinger Report*.

Lucas Nunes Vieira, Minako O'Hagan, and Carol O'Sullivan. 2020. Understanding the societal impacts of machine translation: a critical review of the literature on medical and legal use cases. *Information, Communication & Society*.

Kelsey Percival. 2022. Google's translate tool hardcoded into city website. *Durango Herald*.

Marcelo O. R. Prates, Pedro H. Avelar, and Luís C. Lamb. 2020. Assessing gender bias in machine translation: a case study with Google Translate. *Neural Computing and Applications*, 32(10):6363–6381.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A Neural Framework for MT Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Natália Resende and Andy Way. 2021. Can Google Translate Rewire Your L2 English Processing? *Digital*, 1(1):66–85. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

Aditya Siddhant, Ankur Bapna, Orhan Firat, Yuan Cao, Mia Xu Chen, Isaac Caswell, and Xavier Garcia. 2022. Towards the Next 1000 Languages in Multilingual Machine Translation: Exploring the Synergy Between Supervised and Self-Supervised Learning. ArXiv:2201.03110 [cs].

Elke Teich. 2003. i-iv. In *A methodology for the investigation of translations and comparable texts*, pages i–iv. De Gruyter Mouton, Berlin, Boston.

Arda Tezcan, Joke Daems, and Lieve Macken. 2019. When a 'sport' is a person and other issues for NMT of novels. In *Proceedings of the Qualities of Literary Machine Translation*, pages 40–49. European Association for Machine Translation.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation. ArXiv:1808.10432 [cs].

Gideon Toury. 1995. *Descriptive Translation Studies – and beyond*. John Benjamins Publishing Company. Publication Title: btl.100.

Jonas-Dario Troles and Ute Schmid. 2021. Extending Challenge Sets to Uncover Gender Bias in Machine Translation: Impact of Stereotypical Verbs and Adjectives. In *Proceedings of the Sixth Conference on Machine Translation*, pages 531–541, Online. Association for Computational Linguistics.

Eva Vanmassenhove, Dimitar Shterionov, and Matthew Gwilliam. 2021. Machine Translationese: Effects of Algorithmic Bias on Linguistic Complexity in Machine Translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2203–2213, Online. Association for Computational Linguistics.

Bram Vanroy. 2021. *Syntactic difficulties in translation*. dissertation, Ghent University.

Bram Vanroy, Moritz Schaeffer, and Lieve Macken. 2021. Comparing the Effect of Product-Based Metrics on the Translation Process. *Frontiers in Psychology*, 12.

Martin Volk, Chantal Amrhein, Noëmi Aepli, Mathias Müller, and Phillip Ströbel. 2016. Building a Parallel Corpus on the World's Oldest Banking Magazine. Bochum.

Martin Volk, Noah Bubenhofer, Adrian Althaus, Maya Bangerter, Lenz Furrer, and Beni Ruef. 2010. Challenges in building a multilingual alpine heritage corpus. Malta. University of Zurich.

Rebecca Webster, Margot Fonteyne, Arda Tezcan, Lieve Macken, and Joke Daems. 2020. Gutenberg goes neural : comparing features of Dutch human translations with raw neural machine translation outputs in a corpus of English literary classics. *INFORMATICS-BASEL*, 7(3). Number: 3.

G.U. Yule. 1944. *The statistical study of literary vocabulary*. The University Press. Tex.lccn: 44029835.

Mike Zhang and Antonio Toral. 2019. The Effect of Translationese in Machine Translation Test Sets. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 73–81, Florence, Italy. Association for Computational Linguistics.

Appendix A SacreBLEU scores

corpus	azure	deepl	google	mtX
WMT21	59.0	**69.9**	58.5	53.1
WMT22	50.7	48.5	**52.3**	47.1
Tatoeba	40.6	**42.0**	41.8	39.7
transX	33.0	**36.7**	33.1	32.0
JaneEyre	18.7	**20.1**	19.5	18.6
CSBull	31.5	**32.2**	31.7	30.1
Text+Berg	23.5	**27.2**	24.4	24.6

Table 3: SacreBLEU scores v 2.2.1 across full-sized corpora per MT system. The best values are in **bold**.

Appendix B Lexical richness scores

	system	TTR	Yule's	MTLD		system	TTR	Yule's	MTLD
WMT21	humanA	24.1	**610.79**	134.47	Tatoeba	human	**21.37**	**333.36**	**82.49**
	humanC	23.25	533.61	125.93		azure	20.39	274.15	72.49
	humanD	23.83	562.83	129.73		deepl	20.27	276.48	69.29
	azure	22.84	456.32	129.14		google	20.82	288.72	73.28
	deepl	22.89	475.89	127.37		mtX	20.92	283.91	71.02
	google	23.06	488.02	129.74	transX	human	22.8	**624.76**	**138.32**
	mtX	**24.13**	528.95	**136.34**		azure	22.19	504.05	132.47
						deepl	22	498.44	130.55
WMT22	humanA	19.17	369.49	109.5		google	22.73	548.7	136.45
	humanB	19.76	**405.13**	111.3		mtX	**22.99**	552.01	136.65
	azure	19.25	349.45	113.48	CS Bulletin	human	**7.8**	**69.38**	**279.39**
	deepl	19.3	360.71	110.55		azure	6.55	46.88	249.45
	google	19.7	379.68	112.81		deepl	6.44	44.84	228.18
	mtX	**20.3**	395.08	**117.8**		google	6.88	54.13	258.3
						mtX	6.7	48.75	249.33
Jane Eyre	human	8.08	59.31	126.88	Text+Berg	human	**9.5**	**91.95**	**276.93**
	azure	8.09	54.45	136.64		azure	8.1	58.47	201.06
	deepl	8.15	52.06	127.14		deepl	8.37	67.3	203.15
	google	8.38	**63.31**	**136.87**		google	8.62	70.22	212.04
	mtX	**8.56**	58.01	129.51		mtX	8.07	56.65	191.42

Figure 5: Lexical richness measured with Type-Token Ratio (TTR), reversed Yule's K (Yule's I), and the Measure of Textual Lexical Diversity (MTLD) across all corpora. Higher scores (in **bold**) indicate higher lexical richness.

Appendix C Lexical frequency profile

	system	B1 ↓	B2	B3 ↑		system	B1 ↓	B2	B3 ↑
WMT21	humanA	**72.43**	8.49	**19.08**	Tatoeba	human	**67.72**	6.39	**25.9**
	humanC	73.51	8.27	18.23		azure	69.37	6.51	24.12
	humanD	72.77	8.48	18.75		deepl	70.03	6.41	23.55
	azure	74.11	8.37	17.52		google	68.73	6.52	24.75
	deepl	74.1	8.26	17.64		mtX	68.81	6.55	24.64
	google	73.75	8.41	17.84	transX	human	**78.13**	8.88	**12.99**
	mtX	72.97	8.52	18.51		azure	79.41	8.35	12.24
WMT22	humanA	75.65	7.65	16.7		deepl	79.5	8.31	12.19
	humanB	75.12	7.55	17.33		google	78.76	8.58	12.65
	azure	75.62	7.63	16.75		mtX	78.61	8.52	12.87
	deepl	75.57	7.52	16.91	CS Bulletin	human	**83.34**	7.53	**9.14**
	google	75.1	7.61	17.29		azure	84.2	7.73	8.07
	mtX	**74.6**	7.66	**17.74**		deepl	84.12	7.66	8.22
Jane Eyre	human	79.07	5.95	14.98		google	83.84	7.81	8.35
	azure	79.7	5.65	14.65		mtX	83.85	7.86	8.3
	deepl	80.06	5.48	14.46	Text+Berg	human	**71.55**	6.19	**22.25**
	google	79.5	5.58	14.92		azure	73.61	6.13	20.25
	mtX	**79.02**	5.71	**15.27**		deepl	73.41	6.04	20.55
						google	72.85	6.15	21
						mtX	73.97	5.97	20.06

Figure 6: Lexical frequency profile with B1 indicating top 1000 most frequent words, B2 1000-2000 top frequent words and B3 all the other words.

Appendix D Morphological richness scores

	system	H ↑	D ↓		system	H ↑	D ↓
WMT21	humanA	**85.56**	**47.05**	Tatoeba	human	86.74	47.52
	humanC	83.16	48.41		azure	86.17	47.9
	humanD	84.38	47.82		deepl	87.77	47.59
	azure	82.75	48.32		google	87	47.55
	deepl	83.48	48.1		mtX	**88.29**	**46.93**
	google	83.29	48.11	transX	human	80.21	49.82
	mtX	82.85	48.15		azure	**80.57**	**49.45**
WMT22	humanA	**82.79**	**48.98**		deepl	79.72	49.86
	humanB	82.63	49.02		google	80.14	49.89
	azure	82.3	49.44		mtX	79.22	49.93
	deepl	81.33	50	CS Bulletin	human	82.72	50.38
	google	81.48	49.7		azure	86.12	49.04
	mtX	82.34	49.26		deepl	85.01	49.45
Jane Eyre	human	85.87	48.5		google	85.47	49.33
	azure	86.82	48.06		mtX	**86.25**	**48.98**
	deepl	**87.69**	**47.65**	Text+Berg	human	84.36	49.41
	google	86.46	48.25		azure	**85.79**	49
	mtX	85.9	48.32		deepl	85.69	**48.81**
					google	84.65	49.47
					mtX	84.65	49.46

Figure 7: Morphological richness measured with Shannon entropy (H) and Simpson's diversity (D). Higher H and lower D indicate morphologically richer text (marked in **bold**).

Appendix E Syntactic Equivalence

human or MT	translation
eng	Couple MACED at California dog park
human1	Paar in Hundepark in Kalifornien mit Pfefferspray besprüht
human2	Paar bekommt beim Mittagessen in einem Hundepark Pfefferspray ins Gesicht gesprüht
human3	Angriff mit Pfefferspray auf ein Paar in einem Hundepark in Kalifornien
Online-W	Paar MACED in Kalifornien Hundepark
Online-G	Paar MACED im California Dog Park
nuclear_trans	Paar MACED bei California Dog Park
ICL	Paar MACED bei California Hund Park
VolcTrans-GLAT	Paar MACED in Kalifornien Hundepark
P3AI	Paar Maced im kalifornischen Hundepark
eTranslation	Paar MACED im kalifornischen Hundepark
WeChat-AI	Paar MACED im kalifornischen Hundepark
Manifold	Paar MACED im kalifornischen Hundepark
VNVIDIA-NeMo	Paar MACED im kalifornischen Hundepark
BUPT_rush	Paar MACED im kalifornischen Hundepark
Online-A	Paar MACED im kalifornischen Hundepark
Online-Y	Paar MACED im kalifornischen Hundepark
Online-B	Paar MACED im kalifornischen Hundepark
HuaweiTSC	Paar MACED im kalifornischen Hundepark
UEdin	Paar MACED im kalifornischen Hundepark
UF	Paar MACED im kalifornischen Hundepark
happypoet	Paar MACED im kalifornischen Hundepark
Facebook-AI	Paar MACED im kalifornischen Hundepark
VolcTrans-AT	Paar zerfleischt im kalifornischen Hundepark
Google	Paar MACED im kalifornischen Hundepark
DeepL	Ehepaar wird in kalifornischem Hundepark angegriffen
Azure	Paar MACED im kalifornischen Hundepark
mtX	Paar MACED im kalifornischen Hundepark

Table 4: The first clause of the first sentence in the WMT21 test set in the original English and its German translations, performed by 3 human translators and 20 participating MT systems. The bottom section of the table contains the same clause translated with the commercial MT systems for this paper.

Training and Evaluating Norwegian Sentence Embedding Models

Bernt Ivar Utstøl Nødland
Norwegian Defence Research Establishment
Instituttveien 20
2007 Kjeller
bernt-ivar-utstol.nodland@ffi.no

Abstract

We train and evaluate Norwegian sentence embedding models using the contrastive learning methodology SimCSE. We start from pre-trained Norwegian encoder models and train both unsupervised and supervised models. The models are evaluated on a machine-translated version of semantic textual similarity datasets, as well as binary classification tasks. We show that we can train good Norwegian sentence embedding models, that clearly outperform the pre-trained encoder models, as well as the multilingual mBERT, on the task of sentence similarity.

1 Introduction

Recently there have been a huge increase in the capabilities of natural language processing systems. The new dominant paradigm is using large language models such as BERT (Devlin et al., 2019) or GPT (Radford et al., 2018) as a starting model which one adapts to any given task one wishes to solve. There exists several different versions of BERT-type encoder models in Norwegian (Kummervold et al., 2021), (Kutuzov et al., 2021), (Pyysalo et al., 2021). It is well-known that BERT-type models that give contextual words embeddings do not give particularly good sentence embeddings (Reimers and Gurevych, 2019). For this reason we train and evaluate Norwegian sentence embedding models, using the pre-trained encoder models as starting points.

We train models using the state of the art SimCSE methodology, similarly to the original paper (Gao et al., 2021). Like them, we train both unsupervised and supervised models. We start with a pretrained bidirectional language encoder model such as BERT or RoBERTa (Liu et al., 2019). For the unsupervised version we sample texts from the Norwegian Colossal Corpus (NCC) dataset (Kummervold et al., 2022). We then pass them through the model using two different dropout masks and predict contrastively which pairs within a batch represent the same text. For the supervised version, we train on a machine-translated version of natural language inference (NLI) data, where we use sentences related by "entailment" as positive sentences, and sentences labeled as contradiction as hard negative sentences. We train on both the Norwegian dataset, and a combined dataset of both Norwegian and English NLI data, and show that the latter gives better results for sentence representations in Norwegian. We evaluate our models on a machine translated version of semantic textual similarities (STS) datasets, as well as on the sequence classification problems in Norwegian "Talk of Norway" and the binary classification version of the NoReC review dataset (Velldal et al., 2018).

Our main contributions are:

1. We train and evaluate Norwegian unsupervised and supervised sentence embedding models.

2. We demonstrate a new way to compare the various existing Norwegian language models by measuring their performance after training them to make sentence embeddings.

3. We show that our sentence encoders sometimes get better performance than the base encoder on classification . In particular, we obtain new state of the art results on the classification problem "Talk of Norway".

4. Through our experiments we illustrate the usefulness of machine translated datasets for training and evaluating Norwegian language models. In particular, we show that supervised training on machine translated data out-

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 228–237

performs unsupervised training on Norwegian data.

2 Related work

The fundamental technique we build on is that of training large transformer models (Vaswani et al., 2017). In particular, we utilize the large encoder models Bidirectional Encoder Representations from Transformers (BERT) and Robustly Optimized BERT (RoBERTa) by using them as pre-trained starting points.

Our work builds upon existing language models trained in Norwegian. The National Library of Norway has trained BERT models in Norwegian (Kummervold et al., 2021), which we call NB-BERT, which exists in both base and large size. Also, the language technology group at the University of Oslo has trained their version of a BERT for Norwegian called NorBERT (Kutuzov et al., 2021). There is also a WikiBERT model trained on Norwegian Wikipedia (Pyysalo et al., 2021). We also test the multilingual version of BERT (Devlin et al., 2019), which is trained in Norwegian and many other languages.

Our work uses existing methodology for making sentence embedding models. The first paper to improve BERT to make better sentence representations by training it for that purpose, was the Sentence-BERT paper (Reimers and Gurevych, 2019), which trained sentence embedding models by using siamese networks. We build upon the newer Simple Contrastive learning of Sentence Embeddings (SimCSE) methodology (Gao et al., 2021), which uses a contrastive training objective to create sentence embeddings from a pre-trained encoder. The idea behind both of these works is that of finding a training procedure that better extracts the knowledge about sentences that already exists in the pre-trained encoder model.

Most existing work in the literature on making sentence embeddings are either in English or uses multilingual models. Examples of the latter are mBERT and several other approaches such as (Feng et al., 2022), (Goswami et al., 2021) and (Reimers and Gurevych, 2020).

3 Data

For the unsupervised models, we sample data from the Norwegian Colossal Corpus (NCC) (Kummervold et al., 2022). This is a dataset of different smaller Norwegian text corpuses that has been collected into one corpus by the National Library of Norway to train language models. This is primarily a Norwegian corpus, although there are some amounts of other languages present. The dataset description estimates that 87% of documents are in Norwegian, with about 6-7 % of documents in English and the rest in other European languages (mostly other Nordic languages). We sample 1 million texts from the dataset for training unsupervised. Some are longer than one sentence, but all are truncated to max 32 tokens before training, thus they are all approximately sentence length.

Sentence: Deltakerne mente at hvis interessenter var seriøse om å forbedre finansrapporteringsmodellen, ville en gruppe bli opprettet og finansiert spesielt for dette formålet.
[Translation: Participants believed that if stakeholders were serious about improving the financial reporting model, a group would be created and funded specifically for this purpose.]
Positive: Deltakerne forventer at seriøse interessenter vil danne en gruppe for å forbedre finansrapporteringsmodellen.
[Translation: The participants expect that serious stakeholders will form a group to improve the financial reporting model.]
Negative: A group was created to improve the financial reporting model.

Figure 1: An example of a triplet of sentences of mixed language in the Norwegian/English NLI dataset.

For supervised training we train with data collected for the task of natural language inference (NLI). This task is that of taking a pair of sentences and predicting the relationship between them as either ”entailment”, ”neutral” or ”contradiction”. The authors of the SimCSE paper use NLI data to create triples of a sentence with one positive and one hard negative and show that this data work well for training sentence models using contrastive learning, thus we follow this practice. We use a dataset that has been curated for training in Norwegian by the National Library of Norway.[1] The original data is based on the English datasets the Stanford Natural Language In-

[1] https://huggingface.co/datasets/NbAiLab/mnli-norwegian

Sentence 1: en mann skjærer opp en agurk .
[Translation: a man cuts open a cucumber .]
Sentence 2: en mann skjærer en agurk .
[Translation: a man cuts a cucumber .]
Similarity: 4.2

Sentence 1: en mann spiller harpe .
[Translation: a man plays the harp .]
Sentence 2: en mann spiller et keyboard .
[Translation: a man plays a keyboard .]
Similarity: 1.5

Figure 2: Examples from the translated STS-Benchmark dataset. Similarity ratings are from 0-5.

ference (SNLI) Corpus (Bowman et al., 2015) and Multi-Genre Natural Language Inference (MNLI) dataset (Williams et al., 2018). The Norwegian data is machine translated from the MNLI dataset and has about 128 thousand triples. There is also a combined Norwegian and English version of the dataset made by taking a combination of the translated Norwegian MNLI data and English MNLI and SNLI data.[2] Also included are extra combined Norwegian/English sentence triples: For each of the translated triples there is a joint Norwegian/English triple consisting of one or two sentences in each of English and Norwegian, see Figure 1 for an example. The English/Norwegian dataset contains about 531 thousand triples of sentences.

For evaluation we also machine translate the standard English datasets for semantic textual similarity STS12-16 (Agirre et al., 2012), (Agirre et al., 2013), (Agirre et al., 2014), (Agirre et al., 2015), (Agirre et al., 2016), STSBenchmark (Cer et al., 2017), and SICK relatedness (Marelli et al., 2014). The task is predicting how similar a pair of sentences are to each other on a scale of 0-5. We use these datasets only for validation and testing and never for training. In fig. 2 we see two examples from the translated STS Benchmark dataset.

The usage of translated datasets is a weakness compared to having original data in Norwegian. This project can also be viewed as an exploration of what performance it is possible to get from auto-translated English datasets: To the degree they are shown to be useful, one will have much more data one could potentially work with in Norwegian language processing. We note that for sentence similiarity, a similar exploration of translated data has been done for Swedish in (Isbister and Sahlgren, 2020). They conclude that they do not recommend the usage of automatically translated STS datasets for fine-tuning, but that it should probably have limited negative consequences for comparing models. We partly follow their recommendation: We only use translated STS data for valdiation and evaluation, but we do perform supervised training on translated NLI data.

[2]The same English data that was used to train English SimCSE: https://huggingface.co/datasets/princeton-nlp/datasets-for-simcse

4 Experiments

Our experiments follow the implementations in the SimCSE paper closely. We start with a pretrained encoder model that is either BERT or RoBERTa.

For unsupervised training we sample one million texts from the NCC dataset. We then pass each text through the model using two different dropout masks to obtain two different text representations s_i and s_i^+ for each text. Here dropout functions as a form of continuous augmentation of embeddings. Then we contrastively predict which pairs of texts within a batch are the same using cross-entropy loss on the cosine similarity scores. In other words, the loss for text i is given by

$$\text{loss}_i = -\log \frac{e^{\text{sim}(s_i, s_i^+)/\tau}}{\sum_{j=1}^{b} e^{\text{sim}(s_i, s_j^+)/\tau}},$$

where sim is cosine similarity and τ is a temperature hyperparameter which we simply set to 0.05, which is the outcome of optimization done in the SimCSE paper.

For training unsupervised models, the models we start from are given by their names on huggingface as

- bert-base-cased [english model]
- roberta-base [english model]
- bert-base-multilingual-cased
- TurkuNLP/wikibert-base-no-cased
- ltgoslo/norbert2
- NbAiLab/nb-bert-base

Model	Avg. STS
BERT	34.29
RoBERTa	25.56
mBERT	48.34
WikiBERT	42.21
NorBERT	54.42
NB-BERT-base	50.41
NB-BERT-large	49.90

Table 1: Average performance of models before training using average of the last layer on Norwegian STS.

- NbAiLab/nb-bert-large

The english models are included as a sanity check: Since we are using automatically translated datasets to choose the best models, we want to compare their performance with some models that are expected to perform worse than Norwegian models. For the same reason we also test on the English STS datasets.

We train the supervised models using NLI data where each sentence has one paired sentenced labeled as entailment, which is regarded as a positive sample, and one sentence labeled with contradiction, which is considered a negative sample. We thus obtain three different sentence representations s_i, s_i^+, s_i^-. As in the SimCSE paper, we train contrastively trying to predict the positive pairs, and add the negative sentence representation s_i^- to the loss function as follows:

$$\text{loss}_i = -\log \frac{e^{\text{sim}(s_i, s_i^+)/\tau}}{\sum_{j=1}^{b} e^{\text{sim}(s_i, s_j^+)/\tau} + e^{\text{sim}(s_i, s_j^-)/\tau}} \quad (1)$$

For training supervised models we start with the following models:

- bert-base-multilingual-cased
- TurkuNLP/wikibert-base-no-cased
- ltgoslo/norbert2
- NbAiLab/nb-bert-base
- NbAiLab/nb-bert-large

We train with the same settings as in the SimCSE paper: We set a max sequence length of 32, and use the learning rates and batch sizes given in the appendix of the SimCSE paper (which vary by model type and size). Each model is trained on a single NVIDIA 3090 GPU. For some models we have to use gradient accumulation to achieve the correct batch size due to lack of RAM, which changes training dynamics a bit, since contrastive loss depends on the entire batch. We do not see any noticable effects on results from this. We train with the Adam optimizer with linear weight decay and put a multi-layer perceptron (MLP) on top of the model for training. Unsupervised we train for one epoch, and supervised for three. The best model is selected by evaluating on the dev part of the STS Benchmark dataset. For evaluation we test both with and without this MLP, and find that generally, testing without the MLP gives slightly better results. We train three versions of each model and report average scores.

The models are also fine-tuned on two Norwegian sequence classification tasks. Talk of Norway (ToN) is a subset of the Norwegian parliament speeches dataset (Lapponi et al., 2018), where the task is to classify whether the speech was given by SV or FrP (politically left or right, respectively) selected in (Kummervold et al., 2021).[3] NoReC is a dataset of reviews in Norwegian from different domains such as movies, video games and music (Velldal et al., 2018). From this dataset one can extract a binary classification task by taking the subset of reviews that are clearly positive or negative and letting the task be to classify them as positive or negative (Øvrelid et al., 2020). We take the text representations made by the model before the MLP, and add a linear classification layer on top and fine-tune the entire model on the training dataset. For both the fine-tuning datasets we do a grid search for hyperparameters under the following conditions (these are the same hyperparameters as in the finetuning examples in the appendix of the original BERT paper (Devlin et al., 2019)):

- epochs=2, 3, 4
- learning rate = 2e-5, 3e-5, 5e-5
- batch size 16, 32

We use the macro f1 score on the validation set to select the best model for each training run. We do three training runs and report the average of test scores.

[3] https://huggingface.co/datasets/NbAiLab/norwegian_parliament

Model	STS12	STS13	STS14	STS15	STS16	STSB	SICKR	Avg.
BERT	55.21	49.64	49.29	63.68	54.39	54.67	50.93	53.97
RoBERTa	60.30	59.12	57.15	68.73	64.33	64.04	54.39	61.15
mBERT	60.88	62.31	55.91	70.78	66.80	61.87	57.13	62.24
WikiBERT	63.38	70.21	62.63	74.04	70.90	70.88	62.52	67.79
NorBERT	56.41	65.33	54.32	68.95	68.00	62.40	64.54	62.85
NB-BERT-base	59.40	70.70	57.93	71.87	69.94	69.25	63.98	66.15
NB-BERT-large	**70.45**	**80.80**	**72.79**	**81.53**	**78.41**	**79.35**	**69.18**	**76.07**

(a) Performance of unsupervised models on the Norwegian STS datasets.

Model	STS12	STS13	STS14	STS15	STS16	STSB	SICKR	Avg.
mBERT	73.43	69.09	70.84	81.50	73.82	76.47	72.79	73.99
WikiBERT	73.29	64.48	69.24	80.32	74.51	75.42	69.94	72.45
NorBERT	74.30	70.69	72.09	82.56	76.91	79.33	73.74	75.66
NB-BERT-base	76.31	77.20	75.43	84.47	77.69	82.14	77.97	78.75
NB-BERT-large	**77.07**	**83.65**	**80.28**	**86.24**	**81.87**	**84.37**	**78.44**	**81.70**

(b) Performance on the Norwegian STS datasets of supervised models trained on both Norwegian and English NLI data.

Model	STS12	STS13	STS14	STS15	STS16	STSB	SICKR	Avg.
mBERT	69.28	71.50	69.44	78.12	74.38	71.12	67.70	71.65
WikiBERT	70.14	71.18	71.79	77.56	76.20	74.20	67.32	72.63
NorBERT	70.79	74.46	72.44	80.66	77.73	76.65	71.56	74.90
NB-BERT-base	72.41	79.22	74.67	81.47	77.72	78.49	73.50	76.78
NB-BERT-large	**74.67**	**83.65**	**79.47**	**84.15**	**81.82**	**82.25**	**74.75**	**80.11**

(c) Performance on the Norwegian STS datasets of supervised models trained on Norwegian NLI data.

Table 2: Results of our models tested on the Norwegian STS datasets(Spearman's correlation).

5 Results sentence similarity

We evaluate the trained models on the semantic textual similarity datasets. We evaluate our models both on the Norwegian version of the datasets, and the original English. We report Spearman's correlation for the STS datasets.

5.1 Evaluation in Norwegian

In Table 1 we see the average performance on the Norwegian STS before training using the average of the last layer to compare embeddings. We also tested using the average of first and last layers (giving similar numbers) and using "cls" token (giving worse numbers). Thus we have a baseline to compare how much the models have learned from the training.

In Table 2a we see the performance of our unsupervised models on the Norwegian STS datasets. These are the results when we test without the MLP, which on average performs slightly better than using MLP also for testing.

In Table 2b we see the results from training supervised models on the combination of Norwegian and English NLI data, while Table 2c shows the performance when training on only Norwegian NLI data. We see that training with English included improves performance over merely training in Norwegian for all models.

We see that the supervised models perform much better than the unsupervised ones. This would usually not be surprising, but considering the supervised data is automatically translated and therefore presumably of lower quality than the unsupervised data, it is interesting to note.

5.2 Evaluation in English

In Table 3a we show the results from testing our unsupervised models on the English dataset. In Table 3b we show the results from testing our supervised models trained on the combined English and Norwegian dataset on the English STS data, while Table 3c shows the results for supervised models trained only on Norwegian data.

Since we have automatically translated the STS data, we are unsure how accurate the ground truth labels in Norwegian will be, since there will be examples of sentences where the similarity of the sentences changes because of differing translations. However we think that this should not influ-

Model	STS12	STS13	STS14	STS15	STS16	STSB	SICKR	Avg.
BERT(english)	54.76	70.77	57.39	69.32	69.19	61.66	66.29	64.20
roBERTa(english)	65.26	77.06	67.09	76.88	76.71	75.32	65.60	71.99
mBERT	63.56	73.10	63.95	74.67	73.56	68.58	61.61	68.43
WikiBERT	64.68	77.60	67.04	76.20	76.30	74.63	65.34	71.68
NorBERT	52.96	62.30	54.99	67.45	69.83	63.68	62.40	61.94
NB-BERT-base	56.23	72.06	57.93	68.71	71.09	67.25	61.63	64.99
NB-BERT-large	**72.54**	**83.68**	**76.08**	**83.03**	**81.09**	**81.32**	**68.80**	**78.08**

(a) Performance of unsupervised models on English STS datasets.

Model	STS12	STS13	STS14	STS15	STS16	STSB	SICKR	Avg.
mBERT	76.88	79.69	77.58	84.99	78.52	81.36	77.30	79.47
WikiBERT	72.45	59.56	67.08	80.87	75.21	75.31	74.01	72.07
NorBERT	73.39	69.40	72.65	83.10	77.30	80.48	76.55	76.13
NBBert-base	76.93	78.78	77.76	85.28	80.29	82.96	78.49	80.07
NBBert-large	**78.30**	**85.92**	**81.78**	**87.11**	**83.24**	**85.72**	**79.56**	**83.09**

(b) Performance of supervised models on English STS datasets fine-tuned on both Norwegian and English MNLI.

Model	STS12	STS13	STS14	STS15	STS16	STSB	SICKR	Avg.
mBERT	72.62	79.36	75.84	81.87	79.70	77.48	70.18	76.72
WikiBERT	65.47	65.30	67.40	76.86	73.12	68.91	60.59	68.24
NorBERT	66.90	68.62	69.63	79.35	76.23	73.38	69.66	71.97
NBBert-base	71.57	80.30	76.30	81.55	79.23	78.09	71.12	76.88
NBBert-large	**76.42**	**85.58**	**81.23**	**85.49**	**83.21**	**83.15**	**75.04**	**81.45**

(c) Performance of supervised models on English STS datasets fine-tuned on Norwegian MNLI.

Table 3: Results of our models tested on the English STS datasets(Spearman's correlation).

ence comparisons between different models very much. This is supported by the fact that the internal ranking between models for the Norwegian and the English dataset is the same among the Norwegian unsupervised models. (English models unsurprisingly are higher in the rankings when tested on English)

One of the more interesting findings in this paper is how strong performance our models get on the English STS data. NB-BERT-base was initialized from the mBERT checkpoint which can partly explain this, but not all models was started from a model pre-trained in English. The unsupervised NB-BERT-large achieves a score of 78.08 on English STS. For comparison, the best unsupervised model in the original SimCSE paper, SimCSE-RoBERTa-large, achieved a score of 78.90. Thus we see that we have a model pre-trained on a Norwegian corpus (containg some English), further trained unsupervised in Norwegian, that achieves less than 1% worse score than the best English model, trained in English. This model is also better than the best unsupervised English model in the original SentenceBERT paper. The supervised NB-BERT trained only on Norwegian NLI achieved a score of 81.45, while the version trained on Norwegian and English NLI achieve a score of 83.09. Comparably the supervised original English version SimCSE-BERT-base got a score of 81.57 and SimCSE-RoBERTa-large 83.76. Thus we see that we achieve comparable performance between a supervised Norwegian large BERT and a supervised English base BERT, when testing in English. Our best supervised model is less than 1% away from the best English SimCSE model, although this is less surprising than for the unsupervised models, since we in this case fine-tune our model also on English NLI. We also note that our best supervised model which is trained on only Norwegian is better than the best supervised English model in the Sentence-BERT paper. Thus it does seem like the models learn a lot for performing well at English sentence similarity even though the pre-training is mostly in Norwegian. The strong performance in English of NB-BERT models was already noted in (Kummervold et al., 2021).

To see if we can better understand the

BERT	76.7
RoBERTa	79.8
mBERT	80.2
WikiBERT	83.2
NorBERT	83.9
NB-BERT-base	82.7
NB-BERT-large	89.7

(a) Performance of unsupervised models when fine-tuned on the Talk of Norway dataset.

mBERT	79.3
WikiBERT	82.6
NorBERT	85.7
NB-BERT-base	83.4
NB-BERT-large	89.3

(b) Performance of supervised models trained on Norwegian NLI when fine-tuned on the Talk of Norway dataset.

mBERT	79.2
WikiBERT	81.1
NorBERT	84.9
NB-BERT-base	83.3
NB-BERT-large	89.3

(c) Performance of supervised models trained in on Norwegian and English NLI on the Talk of Norway dataset.

Table 4: Performance of our models on the ToN dataset(F1 score).

above findings, we tested the English supervised SimCSE-RoBERTa-large on Norwegian STS, and achieved only an average score of 54.23. Thus a very good English model scores badly in Norwegian, while a very good Norwegian model scores well in English. This might indicate that the reason the Norwegian models all perform so well in English is that there is enough English in the Norwegian training data (probably including many snippets in the Norwegian parts) that the models learn quite a lot of English.

6 Results classification

We report macro F1 score for the binary classification tasks.

6.1 ToN binary classification

In Table 4a we see the performance of the unsupervised models when fine-tuned on the Talk of Norway dataset. In Table 4b we see the performance of the supervised models trained on Norwegian NLI and then fine-tuned on the ToN dataset, while Table 4c shows the performance when training on both Norwegian and English NLI.

BERT	63.1
RoBERTa	64.4
mBERT	70.3
WikiBERT	77.0
NorBERT	82.0
NB-BERT-base	84.3
NB-BERT-large	87.6

(a) Performance of unsupervised models, fine-tuned on the NoReC binary classification dataset.

mBERT	72.2
WikiBERT	77.9
NorBERT	82.4
NB-BERT-base	85.9
NB-BERT-large	87.0

(b) Performance of supervised models trained on only Norwegian NLI when fine-tuned on the NoReC binary classification dataet.

mBERT	74.4
WikiBERT	77.6
NorBERT	81.0
NB-BERT-base	84.9
NB-BERT-large	87.3

(c) Performance of supervised models trained on Norwegian and English NLI when fine-tuned on the NoReC binary classification dataset.

Table 5: Performance of our models on the NoReC binary classification dataset(F1 score).

We see that training the models to give better sentence embeddings gives some performance gains on this task, compared to fine-tuning the base model: In (Kummervold et al., 2021) it is reported that NB-BERT achieves a score of 81.8, while NorBERT scores 78.2 and mBERT 78.4 on this task. All our numbers are slightly higher.

We see that for this classification task training to make sentence models with English NLI data included did not help: the numbers are very similar with and without it.

6.2 NoReC binary classification

In Table 5a we see the performance of unsupervised models on the NoReC binary classification task. In Table 5b we see the results of supervised models trained on Norwegian NLI, while in Table 5c we see the results of supervised models trained on Norwegian and English NLI.

For this task it is less clear that we get gains from training sentence embedding models: The highest reported number for this task is NB-BERT-base which is reported as 86.4 in (Kummervold

et al., 2021) and 83.9 in (Kutuzov et al., 2021). Our best score for NB-BERT-base is 85.9, which is not better than this. Our best model NB-BERT-large also does not achieve a higher score than about 87%, which is only slightly better than the smaller models. We do not know the reason we get improvements for ToN classification, and not here. The mBERT model do improve with training, but that is not so surprising, since it is not already as strong in Norwegian as most of the other models.

7 Discussion

We believe that our models perform well on the semantic sentence similarity task, even if we do not have any strict comparison since this is the first evalutation of Norwegian sentence embedding models on the STS data. The Norwegian dataset corresponds to the English one, so the scores of English models on English STS and Norwegian models on Norwegian STS should in principle correspond to each other, but because of the extra noise added by the automatic translation we are not surprised that the Norwegian numbers are a bit worse. We see that the models improve a lot compared to before training, and because they perform quite well even for the English STS datasets, we are confident that they have indeed learned something useful in Norwegian.

The supervised models perform better than our unsupervised models even though the supervised models are trained on machine translated data. This shows that machine translated data could be useful for doing NLP in smaller languages, at least for some tasks such as ours. The difference in the numbers we get for unsupervised and supervised training are similar to the ones in the original SimCSE paper. It is a bit unclear to what extent the specific content and language of the training data is important for performing well on STS tasks. For example, one can improve the performance of English SimCSE by training on unrelated image data (Jian et al., 2022). This might be because the task is a form of clustering, and images and text in other languages are structurally similar enough that the models learn something useful.

From doing our experiments we get comparisons of the different Norwegian language models. This is because this method of making sentence embeddings is mostly a way of extracting the knowledge already learned by the models, since the amount of training we do is much smaller than the amount the models already have been pre-trained. An unsuprising conclusion is that the scale of the model is the most important factor in making good language models. NB-BERT-large is the best model by clear margins for all of our evaluations. This conforms to the general tendency in recent NLP that scaling up models is more effective than tailoring data or architecture on a given scale. Next, we find that for binary classification the models NB-BERT-base and NorBERT perform quite similary, while WikiBERT is generally a bit weaker, while all of them clearly outperform mBERT. For sentence similarity we find different rankings among models: Here unsupervised WikiBERT is the second best model, while the supervised version is the weakest of the Norwegian supervised models. Supervised NB-BERT-base is clearly the second best model, while NorBERT performs worse on the STS task.

We see that training sentence embedding models slightly improves performance on the binary classification tasks, but not by much compared with the base models. There is no clear tendency on whether training supervised or unsupervised improves performance on classification more, since the numbers we get are similar in both cases.

Acknowledgements

We are very grateful to Per Egil Kummervold of the National Library of Norway's AI lab for helpful conversations, as well as for sharing the translated MNLI dataset.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Koustava Goswami, Sourav Dutta, Haytham Assem, Theodorus Fransen, and John P. McCrae. 2021. Cross-lingual sentence embedding using multi-task learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9099–9113, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tim Isbister and Magnus Sahlgren. 2020. Why not simply translate? A first swedish evaluation benchmark for semantic similarity. *CoRR*, abs/2009.03116.

Yiran Jian, Chongyang Gao, and Soroush Vosoughi. 2022. Non-linguistic supervision for contrastive learning of sentence embeddings. In *Advances in Neural Information Processing Systems*.

Per Kummervold, Freddy Wetjen, and Javier de la Rosa. 2022. The Norwegian colossal corpus: A text corpus for training large Norwegian language models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3852–3860, Marseille, France. European Language Resources Association.

Per E Kummervold, Javier De la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. 2021. Operationalizing a national digital library: The case for a Norwegian transformer model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 20–29, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-scale contextualised language modelling for Norwegian. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 30–40, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Emanuele Lapponi, Martin G. Søyland, Erik Velldal, and Stephan Oepen. 2018. The Talk of Norway: a richly annotated corpus of the Norwegian parliament, 1998–2016. *Language Resources and Evaluation*, pages 1–21.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In

Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2020. A fine-grained sentiment dataset for Norwegian. In *Proceedings of the 12th Edition of the Language Resources and Evaluation Conference*, Marseille, France, 2020.

Sampo Pyysalo, Jenna Kanerva, Antti Virtanen, and Filip Ginter. 2021. WikiBERT models: Deep transfer learning for many languages. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 1–10, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Alec Radford, Karthik Harasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Erik Velldal, Lilja Øvrelid, Eivind Alexander Bergem, Cathrine Stadsnes, Samia Touileb, and Fredrik Jørgensen. 2018. NoReC: The Norwegian review corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Dozens of Translation Directions or Millions of Shared Parameters? Comparing Two Types of Multilinguality in Modular Machine Translation

Michele Boggia[♠] **Stig-Arne Grönroos**[♠] **Niki Andreas Loppi**[♢] **Timothee Mickus**[♠]
Alessandro Raganato[♡] **Jörg Tiedemann**[♠] **Raúl Vázquez**[♠]
[♠]{firstname.lastname}@helsinki.fi [♢]nloppi@nvidia.com
[♡]alessandro.raganato@unimib.it

Abstract

There are several ways of implementing multilingual NLP systems but little consensus as to whether different approaches exhibit similar effects. Are the trends that we observe when adding more languages the same as those we observe when sharing more parameters? We focus on encoder representations drawn from modular multilingual machine translation systems in an English-centric scenario, and study their quality from multiple aspects: how adequate they are for machine translation, how independent of the source language they are, and what semantic information they convey. Adding translation directions in English-centric scenarios does not conclusively lead to an increase in translation quality. Shared layers increase performance on zero-shot translation pairs and lead to more language-independent representations, but these improvements do not systematically align with more semantically accurate representations, from a monolingual standpoint.

1 Introduction

Multilinguality, within the scope of neural NLP, can mean either ensuring that computations for different languages are homogeneous, or ensuring that models are trained with data coming from different languages. These two definitions are not as equivalent as they might appear: for instance, modular architectures, where some parameters are specific to a single language, can only be conceived as multilingual under the latter definition.

Both of these trends have been explored across multiple works. Machine translation studies have looked into sharing no parameters at all (Escolano et al., 2021; Lyu et al., 2020), sharing across linguistically informed groups (Fan et al., 2021; Purason and Tättar, 2022), sharing only some components across all languages (Dong et al., 2015; Firat et al., 2016; Vázquez et al., 2020; Liao et al., 2021; Zhu et al., 2020; Kong et al., 2021; Blackwood et al., 2018; Sachan and Neubig, 2018; Zhang et al., 2021), and sharing the entire model (Johnson et al., 2017). Concerns about multilinguality have spearheaded research on how to make representations and systems more reliable for typologically and linguistically diverse data (Bojanowski et al., 2017; Adelani et al., 2022), the distinction between multilingual and monolingual representations (Wu and Dredze, 2020), the specificity of massively-multilingual representations (Kudugunta et al., 2019) or the effects of having more diverse data (Arivazhagan et al., 2019; Aharoni et al., 2019; Costa-jussà et al., 2022; Siddhant et al., 2022; Kim et al., 2021; Voita et al., 2019). In this paper, we study whether these different implementations of multilinguality yield qualitatively different types of representations—in other words: *Are the effects of parameter sharing orthogonal to those of adding new languages?*

To broach this question, we make three simplifying assumptions. *First*, we only consider the task of multilingual machine translation—an exhaustive study of the impact of all multilingual NLP tasks is beyond the scope of this paper. Moreover, massively multilingual language models are known to leverage parallel data to enhance semantic abstractions (Hu et al., 2021; Ouyang et al., 2021; Kale et al., 2021). *Second*, we only consider parameter sharing in the last layers of the encoders: we focus on the intermediary representations acquired directly after the encoder and leave decoders for future study. As language selection tokens would compromise the language independence of the representations, this rules out fully shared decoders.

Authors listed alphabetically. Corresponding author: timothee.mickus@helsinki.fi

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 238–247

Third, we focus on an English-centric scenario: i.e., all translation directions seen during training contain English as a source or target language. While such an approach is not without issues (Gu et al., 2019; Zhang et al., 2020), it makes it possible to select translation directions for zero-shot evaluations in a principled manner. Furthermore, most multilingual translation datasets are highly skewed in any case and contain orders of magnitude more English examples (e.g., Costa-jussà et al., 2022).

We conduct our study by testing encoder outputs on three aspects: *task fitness*, *language independence* and *semantic content*. These features have been discussed in earlier literature: probing pretrained language models for semantic content in particular has proven very fecund (e.g., Rogers et al., 2021; Doddapaneni et al., 2021). As for machine translation, these studies are less numerous, although similar aspects have been investigated (Raganato and Tiedemann, 2018). For instance, Kudugunta et al. (2019) study how the learned representations evolve in a multilingual scenario, whereas Vázquez et al. (2020), Raganato et al. (2019) or Mareček et al. (2020) focus on the use of multilingual-MT as a signal for learning language. As we will show, studying representations under different angles is required in order to highlight the differences underpinning distinct implementations of multilinguality.[1]

2 Experimental setup

2.1 Datasets

We focus on datasets derived from the OPUS-100 corpus (Zhang et al., 2020), built by randomly sampling from the OPUS parallel text collection (Tiedemann, 2012). We construct datasets containing 3, 6, 9, 12, 24, 36, 48, 60 and 72 languages other than English and refer to them as `opus-03`, `opus-06`, and so on. To test the impact on the model performance when adding languages, we build the datasets with an incremental approach, so that smaller datasets are systematically contained in the larger ones. Languages are selected so as to maximize the number of available datapoints—for training, zero-shot evaluation and probing—as well as linguistic diversity. See Appendix A for details.

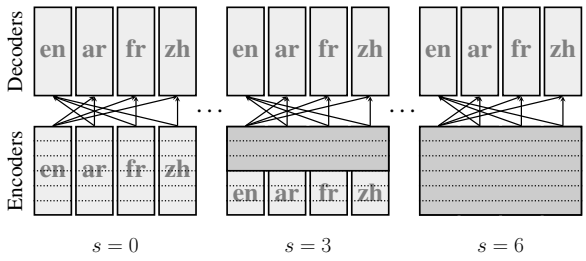


Figure 1: Example model architectures for varying number of shared encoder layers s. Modules with a light grey background are language-specific, modules with a dark grey background are fully shared.

2.2 Models

We train modular sequence-to-sequence Transformer models (Escolano et al., 2021), with 6 layers in the encoder and the decoder. Decoders are systematically language-specific, whereas encoders contain $s \in \{0, \ldots, 6\}$ fully-shared layers on top of $6 - s$ language-specific layers, as shown in Figure 1. We train distinct models for each value of s and each dataset; due to the computational costs incurred, we consider $s \geq 2$ only in combination with datasets up to `opus-12`, as well as `opus-36`. Models vary along two axes: models trained on larger datasets are exposed to more languages, whereas models with higher values of s share more parameters. When training models over a dataset, we consider the translation directions L-to-English, English-to-L, and a L-to-L denoising task, for all languages L in the dataset.[2] The noise model for the denoising auto-encoding objective follows Lewis et al. (2020). An illustration of `opus-03` models is shown in Figure 1. Training details are given in Appendix B

3 Experiments

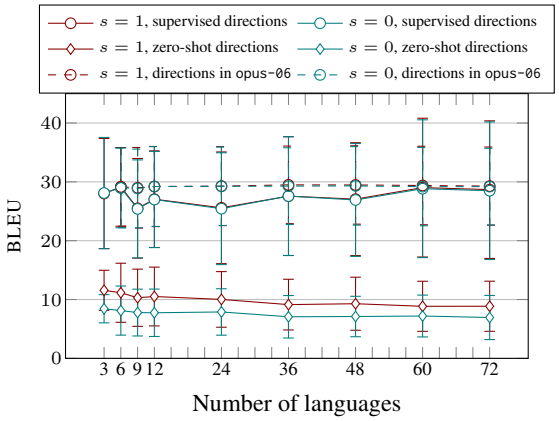
3.1 Task fitness: Machine Translation

The first aspect we consider is the models' performance on machine translation. We report BLEU scores in Figure 2. Where relevant, we also include supervised results for translation directions present in `opus-06` so as to provide comparable scores.[3]

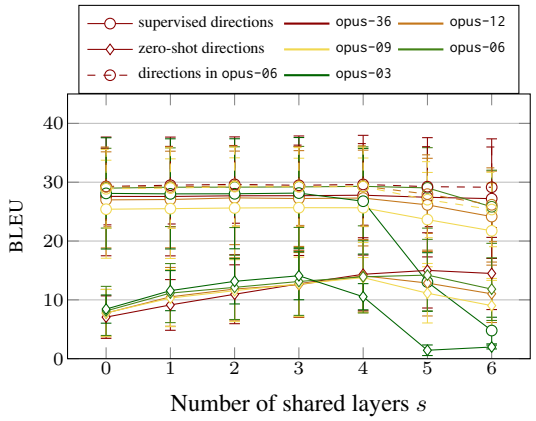
[1] Code available at: `https://github.com/Helsinki-NLP/FoTraNMT/tree/who-would-win`.

[2] I.e., a model trained over the `opus-`n dataset is trained over $3n$ tasks: $2n$ translation tasks, plus n denoising tasks for languages other than English.

[3] Note that all available zero-shot translation directions are systematically present in `opus-06` and all larger datasets.



(a) Average BLEU scores per dataset size



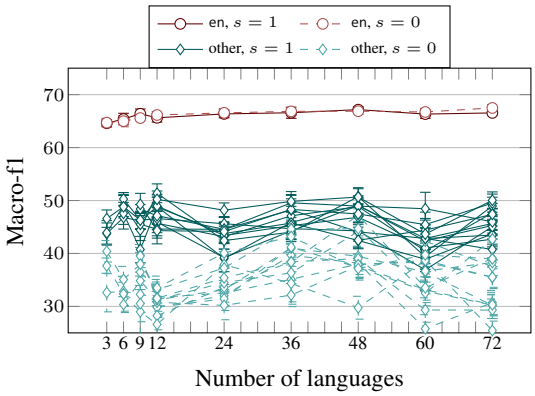
(b) Average BLEU scores per number of shared layers

Figure 2: Average BLEU scores

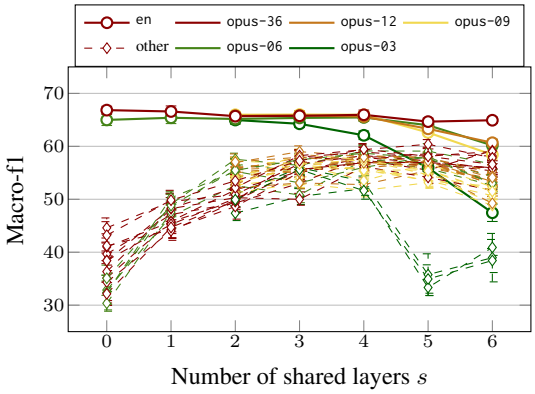
The most obvious trend present is that models trained on opus-03 with $s \geq 5$ underfit, and perform considerably worse than their $s < 5$ counterpart. Otherwise, models with an equivalent number of shared layers s tend to perform very reliably across datasets: e.g., across all supervised translation directions we tested, we found that the maximum variation in BLEU scores for $s < 2$ was of ± 4.8.[4] In Figure 2b, we also observe consistent improvement on zero-shot translation when increasing the number of shared layers s from 0 to 4, and for opus-36 this trend only breaks when the full stack is shared ($s = 6$). Lastly, results in Figure 2a suggest that adding more translation directions decreases zero-shot translation performances, but this trend seems to reverse when a significant number of layers are shared ($s > 3$), as displayed in Figure 2b. In all, under the setup we consider here, it appears that *task fitness and zero-shot generalization are best achieved by sharing more parameters, rather than adding translation directions*—although excessive sharing also impacts performances.[5]

[4] See also Aharoni et al. (2019) or Conneau et al. (2020).

3.2 Language Independence: XNLI



(a) Average XNLI scores per dataset size



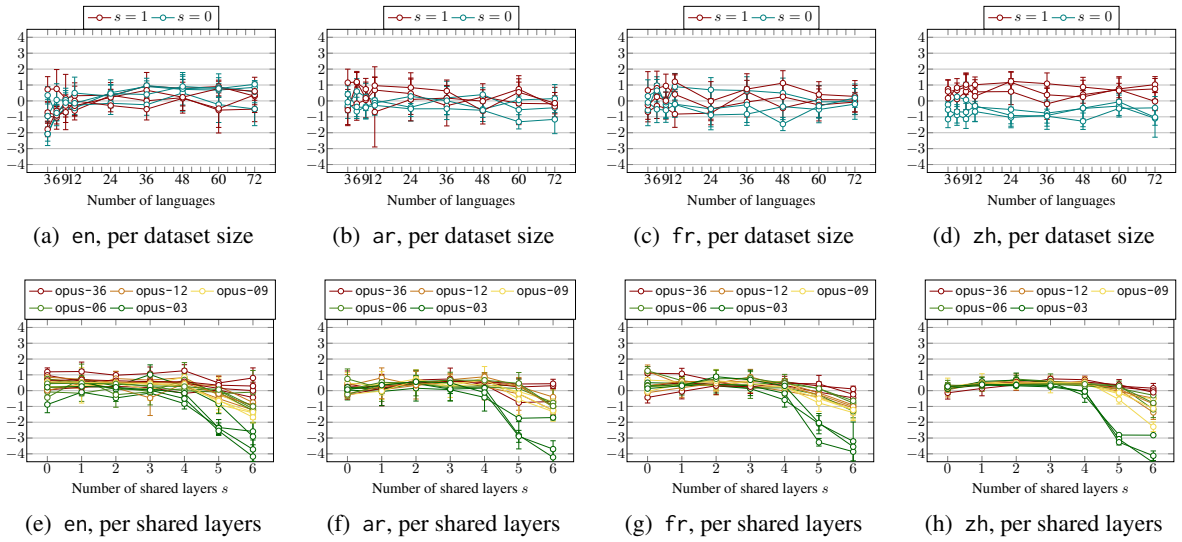
(b) Average XNLI scores per number of shared layers

Figure 3: Average XNLI macro-f1 scores

To test to what degree encoder representations are language-independent, we train classifier probes on XNLI (Conneau et al., 2018). We train models on English and report results for all languages: the gap between English and non-English performances quantifies how language-dependent the representations are. We report macro-f1 on the validation split; if no such split is available, we randomly select 10% instead. See Appendix C for details.

Figure 3 underscores that our English-centric scenario prevents language-independent encoder representations: English targets fare better than their counterparts. Variation seems driven by the number of shared parameters: in Figure 3a, models with $s = 1$ outperform models with $s = 0$, whereas in Figure 3b, higher values of s tend to close the gap between English and other targets. Interestingly,

[5] Previous fully-shared models achieved high zero-shot performances, e.g. Johnson et al. (2017).



(a) en, per dataset size (b) ar, per dataset size (c) fr, per dataset size (d) zh, per dataset size

(e) en, per shared layers (f) ar, per shared layers (g) fr, per shared layers (h) zh, per shared layers

Figure 4: Average macro-f1 scores (z-scaled) on NLU monolingual tasks

higher values of s yield lower f1 scores in smaller datasets, both for English and other languages. In particular, we observe a drop for all languages on opus-03 with $s > 4$, matching the underfitting we saw in Section 3.1; this trend is also attested in all datasets except opus-36. But on the whole, *a greater number of shared parameters leads to more language-independent representations*.

3.3 Semantic Content: NLU benchmarks

To verify the semantic contents captured by our representations, we test them on monolingual GLUE-style benchmarks. We focus on benchmarks for languages present in opus-03: Arabic (ALUE, Seelawi et al. 2021), Chinese (CLUE, Xu et al. 2020), English (GLUE, Wang et al. 2018) and French (FLUE, Le et al. 2020). We select tasks that can be learned using a simple classifier; see Table 4 in Appendix C for a full list of the monolingual classification tasks considered. We follow the same methodology as in Section 3.2.

Results are displayed in Figure 4. Instead of plotting raw macro-f1 scores, we first z-normalize them so as to convert them to a comparable scale. Looking across datasets (Figures 4a to 4d), we do not see a clear variation; at best, we can argue English performances improves when using more language pairs. This is consistent with the English-centric scenario under which we trained our models. Arabic and Chinese results would suggest that $s = 1$ models fare better than $s = 0$ models, but this trend does not carry on convincingly for French.

Comparing across number of shared layers (Figures 4e to 4h) suggests this trend might be more complex: all languages tend to lose in accuracy for higher values of s, and this effect is all the more pronounced for non-English languages and models trained on smaller datasets. For instance, the optimal number of shared layers for Chinese is either $s = 3$ or $s = 4$, depending on the task under consideration and the number of language pairs in the training dataset, but the gain over $s < 3$ models is minimal. This differs crucially from what we observed in Section 3.1, where only $s = 6$ impacted BLEU scores, and in Section 3.2, where there was a clear improvement from low to mid values of s. In sum, probing encoder representations for their semantic contents paints a more nuanced picture, one where *semantic accuracy does not clearly align with task fitness or language-independence*.

4 Conclusions

We have studied whether different means of achieving multilinguality—sharing parameters and multiplying languages—bring about the same effects. What transpires from our experiments is that the two means are not equivalent: we generally observe higher performances and more reliable representations by setting the optimal number of shared parameters. Crucially, this optimum depends on the criteria chosen to evaluate representations: machine translation quality (Section 3.1), language independence (Section 3.2) and semantic accuracy (Section 3.3) all differed in that respect.

These two approaches are not dichotomous: it is possible to both scale the number of languages and

select optimal parameter sharing. What is possible may however not be practical. As guidance to NLP practitioners, we recommend spending effort on tuning the level of parameter sharing for the task at hand. Sharing either too little (0–1 layers in our experiments) or too much (sharing the entire encoder) results in sub-optimal performance overall, but the optimal number of layers to share depends on the task. Spending significant effort on acquiring data for additional language pairs may not yield improved representations past the initial stages of data collection (`opus-03` in our experiments).

Acknowledgements

This work is part of the FoTran project, funded by the European Research Council (ERC) under the EU's Horizon 2020 research and innovation program (agreement № 771113). We also thank the CSC-IT Center for Science Ltd., for computational resources and NVIDIA AI Technology Center (NVAITC) for the expertise in distributed training.

References

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022. A few thousand translations go a long way! leveraging pre-trained models for African news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Graeme Blackwood, Miguel Ballesteros, and Todd Ward. 2018. Multilingual neural machine translation with task-specific attention. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3112–3122, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Sumanth Doddapaneni, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2021. A primer on pretrained multilingual language models. *CoRR*, abs/2107.00676.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Mikel Artetxe. 2021. Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 944–948, Online. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond English-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(1).

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. 2019. Improved zero-shot neural machine translation via ignoring spurious correlations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1258–1268, Florence, Italy. Association for Computational Linguistics.

Junjie Hu, Melvin Johnson, Orhan Firat, Aditya Siddhant, and Graham Neubig. 2021. Explicit alignment objectives for multilingual bidirectional encoders. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3633–3643, Online. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Mihir Kale, Aditya Siddhant, Rami Al-Rfou, Linting Xue, Noah Constant, and Melvin Johnson. 2021. nmT5 - is parallel data still relevant for pre-training massively multilingual language models? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 683–691, Online. Association for Computational Linguistics.

Zae Myung Kim, Laurent Besacier, Vassilina Nikoulina, and Didier Schwab. 2021. Do multilingual neural machine translation models contain language pair specific attention heads? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2832–2841, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Xiang Kong, Adithya Renduchintala, James Cross, Yuqing Tang, Jiatao Gu, and Xian Li. 2021. Multilingual neural machine translation with deep encoder and multiple shallow decoders. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1613–1624, Online. Association for Computational Linguistics.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2020. FlauBERT: Unsupervised language model pre-training for French. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Junwei Liao, Yu Shi, Ming Gong, Linjun Shou, Hong Qu, and Michael Zeng. 2021. Improving zero-shot neural machine translation on language-specific encoders- decoders. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Sungwon Lyu, Bokyung Son, Kichang Yang, and Jaekyoung Bae. 2020. Revisiting Modularized Multilingual NMT to Meet Industrial Demands. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5905–5918, Online. Association for Computational Linguistics.

David Mareček, Hande Celikkanat, Miikka Silfverberg, Vinit Ravishankar, and Jörg Tiedemann. 2020. Are multilingual neural machine translation models better at capturing linguistic features? *The Prague Bulletin of Mathematical Linguistics*.

Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE-M: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 27–38, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Taido Purason and Andre Tättar. 2022. Multilingual neural machine translation with the right amount of sharing. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 91–100, Ghent, Belgium. European Association for Machine Translation.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

Alessandro Raganato, Raúl Vázquez, Mathias Creutz, and Jörg Tiedemann. 2019. An evaluation of language-agnostic inner-attention-based representations in machine translation. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 27–32, Florence, Italy. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Devendra Sachan and Graham Neubig. 2018. Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 261–271, Brussels, Belgium. Association for Computational Linguistics.

Haitham Seelawi, Ibraheem Tuffaha, Mahmoud Gzawi, Wael Farhan, Bashar Talafha, Riham Badawi, Zyad Sober, Oday Al-Dweik, Abed Alhakim Freihat, and Hussein Al-Natsheh. 2021. ALUE: Arabic language understanding evaluation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 173–184, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Aditya Siddhant, Ankur Bapna, Orhan Firat, Yuan Cao, Mia Xu Chen, Isaac Caswell, and Xavier Garcia. 2022. Towards the next 1000 languages in multilingual machine translation: Exploring the synergy between supervised and self-supervised learning. *arXiv preprint arXiv:2201.03110*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Raúl Vázquez, Alessandro Raganato, Mathias Creutz, and Jörg Tiedemann. 2020. A Systematic Study of Inner-Attention-Based Sentence Representations in Multilingual Neural Machine Translation. *Computational Linguistics*, 46(2):387–424.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yilin Yang, Akiko Eriguchi, Alexandre Muzio, Prasad Tadepalli, Stefan Lee, and Hany Hassan. 2021. Improving multilingual translation by representation and gradient regularization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7266–7279, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Changfeng Zhu, Heng Yu, Shanbo Cheng, and Weihua Luo. 2020. Language-aware interlingua for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1650–1655, Online. Association for Computational Linguistics.

A Selected Languages

When constructing larger datasets, we select the additional languages based on four criteria:

(a) maximise the number of datapoints available for training

(b) the presence of zero-shot translation test sets

(c) the existence of XNLI data for the languages

(d) maximize language diversity in the dataset

The information we considered is listed in Table 1, with the exception of criterion (b): only languages in `opus-03` and `opus-06` are relevant to this criterion.

ISO 2	Dataset	Train size	XNLI
ar	opus-03	1,000,000	✓
fr	opus-03	1,000,000	✓
zh	opus-03	1,000,000	✓
de	opus-06	1,000,000	✓
nl	opus-06	1,000,000	✓
ru	opus-06	1,000,000	✓
th	opus-09	1,000,000	✓
tr	opus-09	1,000,000	✓
vi	opus-09	1,000,000	✓
bg	opus-12	1,000,000	✓
el	opus-12	1,000,000	✓
es	opus-12	1,000,000	✓
bn	opus-24	1,000,000	–
eu	opus-24	1,000,000	–
fa	opus-24	1,000,000	–
fi	opus-24	1,000,000	–
he	opus-24	1,000,000	–
id	opus-24	1,000,000	–
it	opus-24	1,000,000	–
ja	opus-24	1,000,000	–
ko	opus-24	1,000,000	–
lv	opus-24	1,000,000	–
mk	opus-24	1,000,000	–
sv	opus-24	1,000,000	–
bs	opus-36	1,000,000	–

(*Continued on next column*)

(*Continued from previous column*)

ISO 2	Dataset	Train size	XNLI
cs	opus-36	1,000,000	–
et	opus-36	1,000,000	–
hu	opus-36	1,000,000	–
is	opus-36	1,000,000	–
lt	opus-36	1,000,000	–
mt	opus-36	1,000,000	–
ro	opus-36	1,000,000	–
sk	opus-36	1,000,000	–
sq	opus-36	1,000,000	–
sr	opus-36	1,000,000	–
uk	opus-36	1,000,000	–
ca	opus-48	1,000,000	–
da	opus-48	1,000,000	–
hr	opus-48	1,000,000	–
mg	opus-48	590,771	–
ml	opus-48	822,746	–
ms	opus-48	1,000,000	–
no	opus-48	1,000,000	–
pl	opus-48	1,000,000	–
pt	opus-48	1,000,000	–
si	opus-48	979,109	–
sl	opus-48	1,000,000	–
ur	opus-48	753,913	–
af	opus-60	275,512	–
cy	opus-60	289,521	–
eo	opus-60	337,106	–
ga	opus-60	289,524	–
gl	opus-60	515,344	–
gu	opus-60	318,306	–
hi	opus-60	534,319	–
ka	opus-60	377,306	–
ne	opus-60	406,381	–
nn	opus-60	486,055	–
sh	opus-60	267,211	–
xh	opus-60	439,671	–
as	opus-72	138,479	–
az	opus-72	262,089	–
br	opus-72	153,447	–
km	opus-72	111,483	–
ku	opus-72	144,844	–
nb	opus-72	142,906	–
pa	opus-72	107,296	–
rw	opus-72	173,823	–
ta	opus-72	227,014	–
tg	opus-72	193,882	–
uz	opus-72	173,157	–
wa	opus-72	104,496	–

(*Continued on next column*)

(*Continued from previous column*)

ISO 2	Dataset	Train size	XNLI

Table 1: Languages selected matched with the first sub-dataset they appear in

B Hyperparameters & Training details

Hyperparameters Models were trained for a total of 100K steps to minimize the negative log-likelihood of the target translation. We accumulate gradients over all translation directions before back-propagation. We optimize our models using AdaFactor (Shazeer and Stern, 2018).

Training occurred on SLURM clusters of A100 NVIDIA GPUs. Each GPU contains the parameters for 3 languages (i.e., 9 translation directions); groups of 4 GPUs form a node. In other words, models for `opus-03` were trained on a single A100 GPU, whereas models for `opus-72` were trained over 24 A100 GPUs, distributed across 6 nodes. We did not go beyond `opus-72` because this matches the largest setup in the computing cluster we used for our experiments. Using the modular training approach with unlimited compute resources, the ideal setup in terms of throughput would contain only one translation direction per GPU as it would allow concurrent training of all translation directions. However, this can introduce larger communication overheads unless all communication calls are not also performed asynchronously and concurrently. A detailed study assessing the training performance, including communication overheads, remains a subject for future work. In this study, all individual models were trained under 36 hours, cf. table 2.

	$s = 0$	$s = 1$
opus03	1 day 08:15:00	1 day 09:46:00
opus06	1 day 03:55:00	1 day 04:08:00
opus09	1 day 04:04:00	1 day 03:44:00
opus12	1 day 04:08:00	1 day 11:25:00
opus24	1 day 04:44:00	1 day 04:37:00
opus36	1 day 05:32:00	1 day 05:43:00
opus48	1 day 05:46:00	1 day 06:30:00
opus60	1 day 05:40:00	1 day 06:02:00
opus72	1 day 05:53:00	1 day 06:21:00

Table 2: Models runtimes

Hyperparameters shared across all models are shown in Table 3; they were set *a priori* so as to not use the validation split of opus-100, as it has been reported to significantly overlap with the test set (Yang et al., 2021). Input data is pre-tokenized using language-specific sentence piece models with 32,000 pieces, except for Chinese and Japanese, where we use 64,000 pieces.

Parameter	Value
src.seq. length	200
tgt.seq. length	200
subword type	sentencepiece
mask ratio	0.2
replace length	1
batch size	4,096
batch type	tokens
normalization	tokens
valid batch size	4,096
max generator batches	2
encoder type	transformer
decoder type	transformer
rnn size	512
word vec size	512
transformer ff	2,048
heads	8
dec layers	6
dropout	0.1
label smoothing	0.1
param init	0.0
param init glorot	true
position encoding	true
valid steps	500,000
warmup steps	10,000
report every	50
save checkpoint steps	25,000
keep checkpoint	3
accum count	1
optim	adafactor
decay method	none
learning rate	3.0
max grad norm	0.0
seed	3435
model type	text

Table 3: Set of hyper-parameters shared across all our models

C Classifiers training procedure

In Sections 3.2 and 3.3, we train classifier probes to investigate the information contained in the encoder spaces. All classifiers correspond to two-layer perceptrons with a hidden layer size of 128, dropout applied to the input layer, and trained with Adam

(Kingma and Ba, 2015) to optimize cross-entropy. We define sentence embeddings by simply taking the sum of the encoder output vectors; the input features of the classifiers are the concatenation of these sentence embeddings. For each set of targets, we train 10 classifiers with different random seeds and report the mean and standard deviation of macro-f1 scores. In Section 3.2, we set the learning rate for XNLI to $5 \cdot 10^{-5}$ with a dropout of $p = 0.1$ and use minibatches of 100 examples. Note that we consider each language in XNLI as a different set of targets, and therefore use different classifiers to compute macro-f1 scores.

	Dataset	Task	Size
ar	NSURL-2019 Task 8	question similarity	10,797
	OSACT4 Task-A	offensive speech detection	6,839
	OSACT4 Task-B	hate speech detection	6,839
en	COLA	linguistic acceptability	8,551
	MRPC	sentence similarity	3,668
	QNLI	NLI	104,743
	QQP	question similarity	363,846
fr	PAWSX	paraphrase detection	49,399
	STSB	paraphrase detection	5,749
	XNLI	NLI	392,702
zh	AFQMC	question similarity	34,334
	CMNLI	NLI	391,783
	TNEWS	news topic classification	53,360

Table 4: NLU monolingual classification tasks

The classification tasks selected for studying the semantic contents of encoder representations in Section 3.3 are shown in Table 4. Due to the limited number of usable tasks in FLUE, we also include a STSB French translation[6] which we binarize by considering similarity judgments > 3 as indicating near-paraphrases. Classifiers discussed in Section 3.3 are trained for 10 epochs with a dropout of 0.1 and a learning rate of $5 \cdot 10^{-5}$, using minibatches of 100 datapoints. We reduced the number of epochs to 5 for all Arabic tasks and used minibatches of 10 examples for the OSACT4 shared tasks A & B due to the longer length of the training examples.

D Limitations

D.1 Material Limitations

As stated in the introduction, we make multiple explicit assumptions that limit the scope of this research. It is plausible that parameter-sharing in the decoder or that replicating our experiments in a non-English-centric scenario will yield a different set of conclusions.

Also worth highlighting are the computational requirements underlying this work: the most demanding experiments require up to 24 A100 NVIDIA GPUs. A side-effect of these demanding computational requirements is that we have not been able to replicate model training across multiple seeds, and therefore report results based on a single model per dataset and number of shared layers. It is also plausible that greatly scaling up the total number of parameters in the networks would affect the conclusions.

Lastly, our use of classifiers to probe for language independence and semantic contents of the representations can be discussed. We have avoided discussing the raw performances of our classifiers, and instead discussed the trends that we observed across our different MT models. Results from our classifiers should be taken as indicators of the aspects we are trying to probe, rather than accurate measures of said aspects: replication studies and further evidence from other settings would be required to establish our models' performances on the criteria we outlined.

D.2 Ethics Considerations

In the present paper, we have argued against adding languages if practical implementation costs are a relevant constraint. We acknowledge that this recommendation may push NLP researchers and engineers towards constructing models specifically for high-resource languages, which would further the coverage gap between low- and high-resource languages.

Nonetheless, it must be stressed that our experiments say nothing of linguistic diversity, as we have ensured that even our smallest dataset (`opus-03`) would contain maximally different languages. Also relevant to the discussion at hand is that one scenario where practical implementation costs are a known constraint is that of developing low-resource languages systems and NLP tools. We believe that providing evidence as to which approach is most effective can prove valuable in such scenarios as well, so as to ensure that efforts can be focused on the most viable path towards endowing lower-resource languages with more efficient and suitable tools.

[6] `https://huggingface.co/datasets/stsb_multi_mt`

DanSumT5: Automatic Abstractive Summarization for Danish

Sara Kolding
School of Communication and Culture
Aarhus University
Jens Chr. Skous Vej 2, 8000 Aarhus C
sarakolding@live.dk

Katrine Nymann
School of Communication and Culture
Aarhus University
Jens Chr. Skous Vej 2, 8000 Aarhus C
katrinesofiemn@hotmail.dk

Ida Bang Hansen
School of Communication and Culture
Aarhus University
Jens Chr. Skous Vej 2, 8000 Aarhus C
idabanghansen@gmail.com

Kenneth C. Enevoldsen
Center for Humanities Computing
Aarhus University
Jens Chr. Skous Vej 4, 8000 Aarhus C
kenneth.enevoldsen@cas.au.dk

Ross Deans Kristensen-McLachlan
Center for Humanities Computing
Aarhus University
Jens Chr. Skous Vej 4, 8000 Aarhus C
rdkm@cas.au.dk

Abstract

Automatic abstractive text summarization is a challenging task in the field of natural language processing. This paper presents a model for domain-specific summarization for Danish news articles. DanSumT5 is an mT5 model fine-tuned on a cleaned subset of the DaNewsroom dataset comprising abstractive article-summary pairs. The resulting state-of-the-art model is evaluated both quantitatively and qualitatively, using ROUGE and BERTScore metrics, along with human rankings of the summaries. We find that although model refinements increase quantitative and qualitative performance, the model is still prone to factual errors. We discuss the limitations of current evaluation methods for automatic abstractive summarization and underline the need for improved metrics and transparency within the field. We suggest that future work should employ techniques for detecting and reducing errors in model output and methods for reference-less evaluation of summaries.

1 Introduction

1.1 Automatic text summarization

Automatic text summarization is the automatic generation of short text which condenses the most salient points of a longer text. Much of the research in this field to date has focused on automatic extractive summarization (El-Kassas et al., 2021), which directly extracts and concatenates sentences from the original text. Various methods have been developed for selecting sentences for extractive summaries. Some, such as TextRank (Mihalcea and Tarau, 2004), rely on measures of sentence importance; others rely on simple heuristics, such as sentence location. For instance, the simple LEAD-3 heuristic selects the first three sentences of a given article (Varab and Schluter, 2020). Extractive summarization thereby ensures grammaticality but tends to suffer from a lack of coherence. Additionally, extractive summaries can be afflicted with dangling anaphoras, in which an extracted sentence refers to a preceding sentence not included in the summary (Gupta and Gupta, 2019).

In recent years, the move towards deep learning in natural language processing (NLP) has brought to the forefront automatic abstractive summarization. Abstractive summaries paraphrase and condense the main points from the original text. This approach views summarization as a text-to-text problem on which models can be trained and fine-tuned. In many cases, abstractive methods enable more informative summarization, since rephrasing and compressing the text allows for more information to be conveyed by fewer sentences. However, factual credibility is not ensured, and the gener-

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 248–264

ated summaries may contain statements that are inconsistent with the original text (Maynez et al., 2020; Zhao et al., 2020).

The majority of existing work on both types of automatic summarization has been in English (Azmi and Al-Thanyyan, 2012; Khan et al., 2019). In this paper, we develop an abstractive text summarization model for Danish news data. We achieve state-of-the-art results by fine-tuning mT5 on a cleaned subset of the DaNewsroom dataset (Varab and Schluter, 2020) consisting of news articles and their corresponding abstractive summaries.

1.2 Previous work

This work builds on the authors' earlier attempts to develop an automatic summarization model for Danish, here referred to as DanSumT5$_{pilot}$. This previous work utilized the DaNewsroom dataset, but with smaller and less thoroughly filtered subsets, and without systematic hyperparameter search. An mT5 model trained on a subset representative of the full dataset performed similarly to extractive baselines validated on the full dataset (Varab and Schluter, 2020). However, upon further inspection, the resulting summaries were predominantly extractive, likely due to the amount of extractive summaries in the dataset. mT5 models trained on more abstractive subsets of the full dataset displayed more qualitatively and quantitatively abstractive behaviour, though the resulting summaries yielded lower quantitative results. The models predominantly generated short and repetitious summaries, possibly resulting in artificially inflated ROUGE performance.

The previous studies yielded two tentative but significant insights. Firstly, we established that fine-tuning an mT5 model capable of producing abstractive summaries does not ensure abstractive summarization. Secondly, our work emphasized the need for employing more nuanced quantitative metrics, as well as qualitative inspection of model output, to determine the quality and abstractiveness of the generated summaries.

The current work expands on these previous efforts by implementing several changes to the data, model, evaluation, and fine-tuning procedure.

1.2.1 Multilingual language models

In this work, we use mT5 (Xue et al., 2021), a multilingual T5 architecture (Raffel et al., 2020) pre-trained on data from 101 languages. There are several reasons for this, beyond the T5 architecture being well-suited to text summarization tasks. While there is evidence to suggest that monolingual models generally perform better on monolingual tasks (Nozza et al., 2020; Popa and Stefănescu, 2020; Rust et al., 2021) (see also section 6), multilingual models seem to increase performance for smaller languages, likely by leveraging cross-lingual transfer (Conneau et al., 2020; Lauscher et al., 2020). Additionally, it has been suggested that this effect depends on the size of the language-specific vocabulary during pre-training, as well as lexical and typological proximity between included languages (Lauscher et al., 2020; Rust et al., 2021). Indeed, target language performance appears to be related to both size of the target-specific pre-training corpora, as well as linguistic similarity between the target and source language (Arivazhagan et al., 2019; Lauscher et al., 2020). We contrast the multilingual architecture with the recently developed monolingual model, DanT5 (Ciosici and Derczynski, 2022).

1.3 Quantitative evaluation metrics

ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-L (R-L) denote the co-occurrence of unigrams (R-1) and bigrams (R-2) in the generated and reference summary (Lin, 2004; Varab and Schluter, 2020), as well as the Longest Common Subsequence (R-L) (Briggs, 2021; Lin, 2004). R-L is thereby the only ROUGE measure capable of considering syntax, since it rewards longer identical sequence overlaps. R-1 and R-2 might inflate performance, even if the generated summary is syntactically incoherent, if numerous co-occurrences from the reference summary are present. ROUGE scores have displayed some correlation with human evaluation of summary fluency and adequacy (Lin and Och, 2004), and are the most commonly used metric for automatic summarization.

It should be noted that ROUGE scores assess lexical overlap of strings, with no reference to semantic similarity. A qualitatively acceptable abstractive summary could consist of completely novel strings, with no overlaps relative to the source text, which would be penalized by the ROUGE metric. Alternative evaluation metrics instead utilize the semantic and syntactic relations captured in contextualised word embeddings produced by transformer-based architectures, such as

BERT (Devlin et al., 2019). BERTScore (Zhang et al., 2020) calculates the similarity between generated and reference summaries as the sum of cosine similarities between the contextual embeddings of tokens. Greedy matching is used to compute a score for each embedding in the generated summary and the most similar embedding in the reference summary. In what follows, we evaluate model performance using a combination of both ROUGE and BERTScore.

2 Dataset

2.1 DaNewsroom dataset

The DaNewsroom dataset (Varab and Schluter, 2020), inspired by the English Newsroom dataset (Grusky et al., 2018), is currently the only publicly available dataset for Danish summarization. The dataset consists of 1.1 million article-summary pairs published over the past 20 years in various Danish news outlets. The summaries were retrieved using a metadata tag and, in most cases, correspond to the subheading of the article.

2.1.1 Data cleaning

To quantify the degree of abstractiveness of a summary, we use the density score (Varab and Schluter, 2020; Grusky et al., 2018), where lower scores indicate more abstractive summaries, and higher scores indicate increasingly more extractive summaries, i.e. summaries containing longer identical sequence overlaps with the original text. Density is defined as:

$$Density(A, S) = \frac{1}{|S|} \sum_{f \in F(A,S)} |f|^2$$

Where *(A,S)* is an instance pair of an article and a summary, and *F(A,S)* is the set of extractive fragments *f* of longest common sequences of tokens in *A* and *S*.

Based on this density measure, the reference summaries in the DaNewsroom dataset are primarily extractive. For the purpose of abstractive summarization, we follow binned density threshold categories (Grusky et al., 2018; Varab and Schluter, 2020), and filter our dataset to contain only abstractive text-summary pairs with a density score between 0 and 1.5.

Several reference summaries in DaNewsroom are short and/or incomplete, as in multiple cases, the web scraping used to collect the dataset extracted incorrect or partial reference summaries. Examples of short or single-word summaries include '2008', 'Et', 'Behandling', and 'P'. Similarly, the dataset also contains some extremely short and/or incomplete articles. In particular, a number of the articles are just one-liners about television scheduling or a paywall:

- Der er lukket for nye kommentarer til denne artikel (*This article is closed for new comments*)
- Svanerne i Slotsparken har fået fem unger (*The swans in Slotsparken have had five cygnets.*)
- 'DR1 — Tirsdag d. 26. august kl. 20:00 - 20:45'. (*DR1 — Tuesday 26 August at 20:00 - 20:45*)

On the other hand, some of the articles are very long. In some cases, the web scraping concatenated the article with a long thread of comments, resulting in articles consisting of several thousand tokens, including English content. We further cleaned the dataset by filtering the summaries and articles based on token length, with lower and upper cutoffs defined respectively as the 2nd and 98th percentiles. Additionally, the data was standardized using ftfy (Speer, 2019) and filtered using various heuristic quality filters, including filters for removing repetitious text, text with a high ratio of non-alphabetic tokens, and text with less than two stop words. This filtering was performed using an implementation similar to textdescriptives (Hansen and Enevoldsen, 2023) and follows an approach similar to Rae et al. (2022). The resulting cleaned subset contained 258,146 abstractive article-summary pairs. Figure 1 shows the distribution of article and summary token lengths before and after this filtering procedure.

3 Model

3.1 Infrastructure

We fine-tuned all of our models using the Transformers library (Wolf et al., 2020) and PyTorch (Paszke et al., 2019) as the back end on 4 RTX8000 GPUs. We used Weights and Biases (Biewald, 2020) for experiment tracking and visualizations.

3.2 Training and models specifications

For our hyperparameter tuning, we trained for 1800 steps with a batch size of 120, and validated

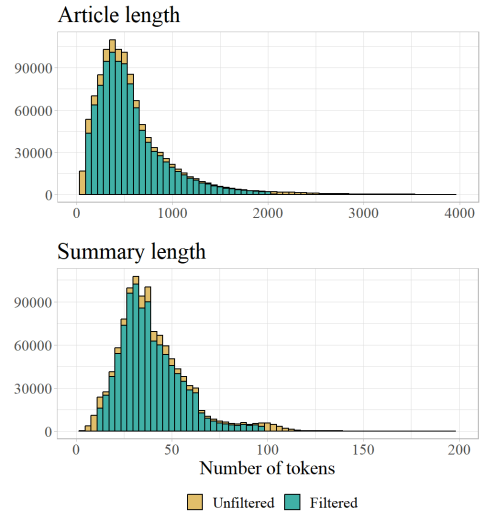


Figure 1: Distribution of summary and article token lengths before and after data cleaning.

with the same batch size, using the full validation set. All models were trained on our filtered dataset using an 80-10-10 split. Due to computing constraints, the hyperparameter search was performed using small-sized models (300M parameters) and for a limited number of steps. The hyperparameter search showed only a few consistent trends and based on this we chose a learning rate of 3.0×10^{-4}, a dropout rate of 0.01, and a polynomial learning rate schedule. For information about the hyperparameter search, see Appendix A[1].

Articles and reference summaries were truncated to a maximum length of 1024 and 128 tokens, respectively. Mixed precision training was employed to lower the memory impact during training. The models were trained using the AdamW optimiser (Kingma and Ba, 2017) with a polynomial learning rate schedule using a learning rate of 3.0×10^{-4}, 2000 warmup steps, and a realized batch size of 16 (a batch size of 8 with an accumulation step of 2). Due to memory constraints, a smaller batch size was used for training as compared to hyperparameter tuning. The models were trained for ten epochs with a dropout rate of 0.01.

For decoding, we used beam search with two beams and a repetitive 3-gram penalty. Additionally, to encourage longer summaries, we set minimum and maximum generation lengths of 9 and 128 and employed a length penalty of 5. The best model was determined according to the cross entropy loss, and the resulting model was tested using a held-out test set of abstractive reference summaries, according to binned density. From this sample of 25,830 article-summary pairs, generated summaries were evaluated by calculating mean density and mean F1-scores for R-1, R-2, R-L (Lin, 2004), and BERTScore (Zhang et al., 2020). The large XLM-RoBERTa model was used to create the embeddings for the BERTScore results (Conneau et al., 2020).

The mT5 model has been made available in different sizes, with larger sizes generally leading to improved performance (Xue et al., 2021). We fine-tuned three different mT5 model sizes; small, base and large, as well as a small DanT5 for comparison. All other parameters were kept constant.

Additionally, we validated the performance of the LEAD-3 and TextRank approaches on our test set. Importantly, both of these methods are extractive, and thus are not necessarily meaningful comparative baselines for abstractive summarization. Still, they provide a benchmark for quantitative comparison, though it should be noted that the test set comprises abstractive reference summaries. We also include performance metrics for DanSumT5$_{pilot}$, which was the best performing abstractive summarizer from our previous unpublished work.

4 Results

4.1 Quantitative results

Table 1 shows R-1, R-2, R-L, and BERTScore results for our three fine-tuned mT5 models and the DanT5 model. Additionally, we present a version of our previous work, DanSumT5$_{pilot}$, trained and evaluated on the same data splits of our current work. We compare these results to the performance of two extractive baseline models, LEAD-3 and TextRank, on the test set.

Mean F1 scores for all metrics are reported. Additionally, 95% confidence intervals for all metrics are calculated using bootstrap resampling with 1000 samples, following the original ROUGE Perl implementation (Li, 2020).

Table 2 shows the mean F1 density scores for the aforementioned models. 95% confidence intervals are calculated using bootstrap resampling with 1000 samples.

The best performing model according to quantitative metrics is DanSumT5$_{large}$. Furthermore, this model generates the lowest-density sum-

[1]Or see the Weight and Biases dashboard at `https://tinyurl.com/3zfuf6vx`

Model	R-1	R-2	R-L	BERTScore
LEAD-3	18.31 [18.21, 18.42]	4.60 [4.55, 4.66]	12.31 [12.25, 12.39]	86.77 [86.75, 86.79]
TextRank	14.80 [14.71, 14.89]	2.82 [2.78, 2.87]	10.03 [9.98, 10.09]	85.86 [85.84, 85.88]
DanT5$_{small}$	20.68 [20.54, 20.82]	5.92 [5.83, 6.02]	15.55 [15.44, 15.67]	88.06 [88.04, 88.09]
DanSumT5$_{pilot}$	19.74 [19.57, 19.90]	6.63 [6.52, 6.74]	16.71 [16.57, 16.85]	88.02 [87.99, 88.05]
DanSumT5$_{small}$	21.42 [21.26, 21.55]	6.21 [6.11, 6.30]	16.10 [15.98, 16.22]	88.28 [88.26, 88.31]
DanSumT5$_{base}$	23.21 [23.06, 23.36]	7.12 [7.00, 7.22]	17.64 [17.50, 17.79]	88.77 [88.74, 88.80]
DanSumT5$_{large}$	**23.76 [23.60, 23.91]**	**7.46 [7.35, 7.59]**	**18.25 [18.12, 18.39]**	**88.97 [88.95, 89.00]**

Table 1: Mean F1 ROUGE and BERTScore performance by model with 95% bootstrapped confidence intervals. Best score is highlighted in bold. Abstractive and extractive methods are delineated.

Model	Density
LEAD-3	26.01 [25.84, 26.18]
TextRank	32.23 [32.07, 32.40]
DanT5$_{small}$	2.91 [2.89, 2.93]
DanSumT5$_{pilot}$	2.76 [2.74, 2.78]
DanSumT5$_{small}$	2.66 [2.65, 2.68]
DanSumT5$_{base}$	2.32 [2.30, 2.34]
DanSumT5$_{large}$	1.91 [1.90, 1.93]

Table 2: Mean F1 density of generated summaries for the different models with 95% bootstrapped confidence intervals. Summaries with a density below 1.5 are considered abstractive, while summaries with a density above 8.19 are considered extractive.

maries, which indicates more abstractive summaries.

4.2 Human evaluation

To manually evaluate the models, we randomly sampled 100 articles in the test set, along with both reference and generated summaries. Two of the authors were selected as raters and tasked with reading the articles and ranking the four summaries (DanSumT5$_{small}$, DanSumT5$_{base}$, DanSumT5$_{large}$, and the reference summary) according to preference[2]. Both raters are women educated to master's level, and both are native speakers of Danish. The rating was blind, with ratings being unaware of the origins of the summaries. The raters were asked to rate based on preference. This rating procedure does not quantify the objective quality of each summary but instead evaluates the relative quality between summaries. Consequently, we cannot determine

[2]See GitHub for the full ratings. Located in the data folder `https://github.com/Danish-summarisation/DanSum.git`

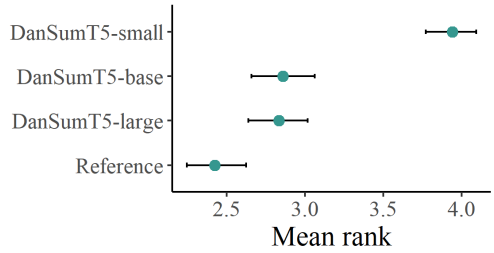


Figure 2: The mean rank obtained for each model through human evaluation. Error bars display the 95% bootstrapped confidence intervals.

why the summary was preferred, or which aspect of the summary contributed to the decision, such as grammaticality or factuality. The two raters had an agreement of 74.8% (95% CI: [71.2, 78.2]). The agreements are calculated based on each pair of comparisons, i.e. every possible comparative relation between summaries. Both raters generally preferred the reference summaries over the generated summaries, as shown in Figure 2.

The best performing model according to subjective evaluation was DanSumT5$_{large}$. The generated summaries are generally grammatically correct and cover the main content of the article, though they tend to suffer from factual inconsistencies (See Appendix B for five randomly chosen examples of reference and model-generated summaries).

5 Discussion

5.1 Evaluation

As seen in Table 1, DanSumT5$_{large}$ achieves higher ROUGE scores and BERTScores than our previous work DanSumT5$_{pilot}$. We have thus achieved state-of-the-art results for Danish abstractive summarization. Furthermore, DanSumT5$_{large}$ generates relatively low-density

summaries. Notably, the mean density falls just outside the abstractive binned density category. Still, the generated summaries are of lower density than those generated by the other models, or by the extractive comparisons. All DanSumT5 models outperform extractive baselines, likely at least partly due to the test set comprising only abstractive reference summaries.

Human evaluation reveals that DanSumT5$_{large}$ generally creates summaries that were highly rated relative to the other summaries. For instance, the DanSumT5$_{large}$ summaries were preferred over the reference summaries in 21.43% of cases. One limitation of summaries generated by all DanSumT5 models is that many of them suffer from factual inconsistencies, as illustrated in Appendix B, Table 7 where DanSumT5$_{small}$ states that artists were hit by a snowstorm, though in actuality, an artificial snowstorm was part of their performance.

5.2 Limitations and Future Directions

5.2.1 ROUGE and BERTScore

Given the inherent limitations of ROUGE scores for evaluating abstractive summaries, we opted to employ BERTScore which does not penalize lexical diversity. Since BERTScore is fully differentiable, it could be used to compute a loss metric for optimisation of both training and evaluation (Zhang et al., 2020). However, some limitations of ROUGE also apply to BERTScore, inasmuch as precision, recall and F1 scores depend on the length of the generated and reference summary. For both ROUGE and BERT metrics, generating very short summaries or using very short reference summaries might inflate performance, since co-occurrences in short sequences are disproportionately rewarded for high relative overlap between reference and model output. Additionally, a high BERTScore is also not a guarantee of factual or grammatical consistency.

Many quantitative evaluation metrics reward similarity with the reference summary; however, this might not be optimal. For instance, since many reference summaries in the DaNewsroom dataset correspond to isolated article subheadings with dangling anaphoras, these issues could transfer to generated summaries. A possible way to remedy this would be to utilize anaphoric information, for instance by checking co-references of the generated summary, and locating errors relating to anaphoric resolution (Steinberger et al., 2007; Sukthanker et al., 2020). Also, despite additional filtering, some reference summaries and articles in the dataset were still incomplete. Mismatched or incomplete summary-reference pairs complicate the task of the model, leading to nonsensical or unrelated outputs: In Appendix B, Table 6, the model-generated summaries contain factual errors and nonsensical phrases, while the reference summary appears unrelated to the accompanying article (full article not included due to copyright). This is because the article was incompletely sampled, whereby critical information was omitted. In these cases, comparisons between generated and reference summaries are illogical. Alternative quantitative metrics suggest omitting the reference summary and evaluating performance using only the original text, for instance by calculating the increase in task performance gained by access to the generated summary (Vasilyev et al., 2020).

Recent research shows that most quantitative metrics do not correlate well with human evaluations of generated summaries in important dimensions such as coherence, consistency, fluency, and relevance (Fabbri et al., 2021; Liu et al., 2017). Indeed, it has also been argued that there is no best practice for reliable human evaluation of summaries, and that human evaluations often do not correlate with other human evaluations of the same summaries (Fabbri et al., 2021; Iskender et al., 2021). Though human evaluations are often presented as the gold standard, evaluator demographics, expertise, and task design hugely affect human evaluations (Harman and Over, 2004; Louis and Nenkova, 2013). Different summaries may focus on different aspects of the same article, with no way to objectively conclude which is better. Consequently, future research might benefit from optimizing reference-free metrics to evaluate generated summaries independent of "gold-standard" counterparts.

5.2.2 Quantifying abstractiveness

There is no clear definition of what counts as an 'abstractive summarization model'. Many studies on abstractive summarization do not report or evaluate the density of the summaries in their dataset, or of their model-generated summaries. A high ROUGE score could correspond to a predominantly extractive or simply very short reference or generated summary, thereby inflating ROUGE performance. We note that our model-generated

summaries have a wide range of density scores, with the largest model producing more abstractive (low-density) summaries. Since abstractive summarization allows high lexical diversity, it is not likely to achieve as high ROUGE scores as extractive summarizers, and thereby low ROUGE performance need not be strongly indicative of poor abstractive summarization.

Finally, existing work tends not to present translated examples of model output. While translation might not be the optimal reflection of non-English summarization, it allows readers to evaluate an approximation of the qualitative results. Lack of transparency, therefore, makes it extremely challenging to evaluate whether the reported summarizers are truly abstractive, or whether high ROUGE performance reflects extractive, short or repetitive summaries.

5.3 Model limitations

5.3.1 Factors limiting practical implementation

Automatic summarization requires the model to be factual to the source text, especially for real-world practical implementations. However, none of the evaluated metrics considers the factual correctness of a generated summary (Falke et al., 2019) and does not reward the model for being factually faithful (Maynez et al., 2020).

Since DanSumT5 sometimes generates summaries with obviously incorrect content (see Appendix B), it is unsuitable for practical implementations where factual accuracy is important, such as summaries of news articles. One possible solution could be to use a separate system to detect such errors in the generated summaries (Falke et al., 2019), such as already existing systems for detection of errors related to quantities (Zhao et al., 2020). Fine-tuning according to this metric, or even using it for reinforcement learning, could alleviate concerns around accuracy and quality. Another approach to enhancing factual accuracy uses question asking to evaluate factual consistency by checking if the generated summary and article yield the same answer (Wang et al., 2020). Future work could extend this approach to Danish.

5.3.2 Data limitations and considerations

Many of the reference summaries suffer from dangling anaphoras since they are scraped from the article's subheading, often lacking the context of the title. These were likely not intended to be read as summaries, or even read in isolation from the article's title. This underlines the importance of data quality, since it defines the upper limit for model performance. We found that most of the 100 DanSumT5 summaries inspected for evaluation avoid dangling anaphoras, likely due to only a minority of the dataset suffering from this artefact, and could thus be argued to be better than the reference summaries in this aspect. For example, in the Appendix B, Table 5 the reference summary refers to "the superstar", while the model summaries mention the actor by name.

The current paper demonstrates that it is possible to fine-tune a multilingual model to create a performant text summarization model for a specific domain of Danish language. Other directions for future work must therefore include further experimentation with different datasets. The practical costs of creating high-quality datasets for this task are a challenge for a language such as Danish which is relatively model-rich, compared to similarly sized languages, and data-poor in terms of high-quality data, compared to larger languages. However, as shown in this paper, good results can be obtained by fine-tuning a multilingual model on a web-scraped dataset with minimal data cleaning.

6 Negative Results

During the training of these models, we attempted a few additional ideas, most of which were shown to be unpromising. This section briefly describes these attempts:

1. As suggested by (Abdaoui et al., 2020) we reduced the model by restricting the vocabulary to Danish and English tokens. Meaningful tokens were estimated using a filtered version of the Danish Gigaword (Strømberg-Derczynski et al., 2021)[3] and English Gigaword (Graff and Cieri, 2003). Reducing the model size allowed us to train these models with a larger batch size. While the reduced size led to similar performance to the original architecture for the small and base-sized models, the large pruned model proved highly unstable during training.

2. During the early phases of development, we also experimented with the recently released

[3]More information about the specific dataset can be found at `https://huggingface.co/datasets/DDSC/dagw_reddit_filtered_v1.0.0`

Danish T5 model, DanT5 (Ciosici and Derczynski, 2022). As part of the initial grid search over hyperparameters, we discovered that this monolingual model performed consistently worse compared to the similarly sized multilingual mT5 models. Table 1 illustrates this disparity on the full dataset. This could be due to DanT5's novel warm-starting approach which utilises an English T5 model checkpoint, but further experimentation is required in this area.

3. This work seeks to train models for abstractive summarization and thus filters out extractive summaries from the training data based on a density threshold. We experimented with lowering the density threshold for the training set summaries to include increasingly more extractive summaries. While we obtained a lower loss when including moderately more extractive references, the resulting summaries were notably more extractive.

4. In an attempt to avoid potential overfitting on the filtered dataset, we also experimented with training a large mT5 model for only 1 epoch using similar hyperparameters with the exception of a dropout rate of 0.1. In the human evaluation, this model placed slightly below the base-sized model of DanSumT5.

7 Conclusion

This paper presents DanSumT5, a set of models achieving state-of-the-art results in automatic abstractive summarization for Danish news articles. These results were achieved by fine-tuning mT5 models and implementing more thorough cleaning of the DaNewsroom dataset. We present state-of-the-art ROUGE and BERTScore performance for Danish abstractive summarization with our DanSumT5$_{large}$. Human inspection of the relative quality of the generated summaries revealed that they were generally grammatical and coherent. We discuss several limitations of the quantitative metrics, emphasizing that ROUGE penalizes lexical diversity inherent to abstractive summarization, while high quantitative performance could obscure low qualitative performance. This emphasizes the need for more transparency in the field, and we argue that research should include more nuanced metrics, as well as manual evaluation of the density and overall quality of the model output. Limitations of our work include data quality and the prevalence of factual errors in the generated summaries. All code related to this project is open-sourced via Github[4], and the model is made freely available via Huggingface for public use[5].

References

Amine Abdaoui, Camille Pradel, and Grégoire Sigel. 2020. Load What You Need: Smaller Versions of Multilingual BERT. *arXiv preprint arXiv:2010.05609*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges. Technical Report arXiv:1907.05019, arXiv.

Aqil M Azmi and Suha Al-Thanyyan. 2012. A text summarizer for Arabic. *Computer speech & language*, 26(4):260–273. Place: Kidlington Publisher: Elsevier Ltd.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

James Briggs. 2021. The Ultimate Performance Metric in NLP.

Manuel R. Ciosici and Leon Derczynski. 2022. Training a T5 Using Lab-sized Resources. ArXiv:2208.12097 [cs].

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. *arXiv:1911.02116 [cs]*. ArXiv: 1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv:1810.04805 [cs].

Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating

[4] https://github.com/Danish-summarisation/DanSum

[5] https://huggingface.co/Danish-summarisation

Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking Generated Summaries by Correctness: An Interesting but Challenging Application for Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

David Graff and Christopher Cieri. 2003. English Gigaword. Artwork Size: 4089446 KB Pages: 4089446 KB Type: dataset.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *arXiv preprint arXiv:1804.11283*.

Som Gupta and S. K Gupta. 2019. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65.

Lasse Hansen and Kenneth Enevoldsen. 2023. TextDescriptives: A Python package for calculating a large variety of statistics from text. ArXiv:2301.02057 [cs].

Donna Harman and Paul Over. 2004. The Effects of Human Variation in DUC Summarization Evaluation. In *Text Summarization Branches Out*, pages 10–17, Barcelona, Spain. Association for Computational Linguistics.

Neslihan Iskender, Tim Polzehl, and Sebastian Möller. 2021. Reliability of Human Evaluation for Text Summarization: Lessons Learned and Challenges Ahead. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 86–96, Online. Association for Computational Linguistics.

Rahim Khan, Yurong Qian, and Sajid Naeem. 2019. Extractive based Text Summarization Using KMeans and TF-IDF. *International Journal of Information Engineering and Electronic Business*, 11:33–44.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. ArXiv:1412.6980 [cs].

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From Zero to Hero: On the Limitations of Zero-Shot Cross-Lingual Transfer with Multilingual Transformers. Technical Report arXiv:2005.00633, arXiv. ArXiv:2005.00633 [cs] type: article.

Jiahao Li. 2020. ROUGE Metric. Accessed 2023-04-16 15:36:28.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04*, pages 605–es, Barcelona, Spain. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian V. Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2017. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. Technical Report arXiv:1603.08023, arXiv. ArXiv:1603.08023 [cs] type: article.

Annie Louis and Ani Nenkova. 2013. Automatically Assessing Machine Summary Content Without a Gold Standard. *Computational Linguistics*, 39(2):267–300.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. *arXiv:2005.00661 [cs]*. ArXiv: 2005.00661.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [MASK]? Making Sense of Language-Specific BERT Models. Technical Report arXiv:2003.02912, arXiv. ArXiv:2003.02912 [cs] type: article.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Cristian Popa and Vlad Stefănescu. 2020. Applying Multilingual and Monolingual Transformer-Based Models for Dialect Identification. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 193–201, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John

Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2022. Scaling Language Models: Methods, Analysis & Insights from Training Gopher. ArXiv:2112.11446 [cs].

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv:1910.10683 [cs, stat]*. ArXiv: 1910.10683.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models. Technical Report arXiv:2012.15613, arXiv. ArXiv:2012.15613 [cs] type: article.

Robyn Speer. 2019. ftfy. Zenodo. Version 5.5.

Josef Steinberger, Massimo Poesio, Mijail Kabadjov, and Karel Jezek. 2007. Two uses of anaphora resolution in summarization. *Information Processing and Management*, 43:1663–1680.

Leon Strømberg-Derczynski, Manuel Ciosici, Rebekah Baglini, Morten H. Christiansen, Jacob Aarup Dalsgaard, Riccardo Fusaroli, Peter Juel Henrichsen, Rasmus Hvingelby, Andreas Kirkedal, Alex Speed Kjeldsen, Claus Ladefoged, Finn Årup Nielsen, Jens Madsen, Malte Lau Petersen, Jonathan Hvithamar Rystrøm, and Daniel Varab. 2021. The Danish Gigaword corpus. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 413–421, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. 2020. Anaphora and coreference resolution: A review. *Information Fusion*, 59:139–162.

Daniel Varab and Natalie Schluter. 2020. DaNewsroom: A Large-scale Danish Summarisation Dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6731–6739, Marseille, France. European Language Resources Association.

Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the BLANC: Human-free quality estimation of document summaries. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 11–20, Online. Association for Computational Linguistics.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and Answering Questions to Evaluate the Factual Consistency of Summaries. Technical Report arXiv:2004.04228, arXiv. ArXiv:2004.04228 [cs] type: article.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771 [cs]*. ArXiv: 1910.03771.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv:2010.11934 [cs]*. ArXiv: 2010.11934.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. Technical Report arXiv:1904.09675, arXiv. ArXiv:1904.09675 [cs] type: article.

Zheng Zhao, Shay B. Cohen, and Bonnie Webber. 2020. Reducing Quantity Hallucinations in Abstractive Summarization. *arXiv:2009.13312 [cs]*. ArXiv: 2009.13312.

Appendices

A Hyperparameters appendix

Table 3 shows the hyperparameter search space. For sampling, we use the Tree-structured Parzen Estimator implemented in Optuna. The search showed only a few consistent trends, and based on this, we chose a learning rate of 3.0×10^{-4}, a dropout rate of 0.01, a polynomial learning rate, and the mT5 model set.

For more information, see the Weights and Biases dashboard at `https://tinyurl.com/3zfuf6vx`.

Models	DanT5$_{small}$, mT5$_{small}$
Learning Rate	0-0.1
Learning Rate Schedule	Polynomial, Linear, Cosine, Constant
No Repeat N-Gram Size	3-5
Length Penalty	4-7
Max Grad Norm	1-10
Data Quality	{True, False}

Table 3: Hyperparameters for Model Search.

B Examples of generated summaries appendix

This appendix consists of five randomly sampled examples of automated summaries generated by our models, along with the corresponding reference summary from the DaNewsroom dataset.

Origin	Summary
Reference	Viborg-Esbjerg 0-5 Et hattrick fra den tidligere OB-spiller Rajko Lekic sender Esbjerg på førstepladsen i Superligaen. *Viborg-Esbjerg 0-5 A hat-trick from former OB player Rajko Lekic sends Esbjerg to first place in the Super League.*
DanSumT5$_{large}$	Esbjerg fører nu Superligaen efter to sejre på stribe og en 5-0-sejr over Viborg på egen bane. *Esbjerg now leads the Super League after two wins in a row and a 5-0 win over Viborg at home.*
DanSumT5$_{base}$	Esbjerg satte Viborg på plads med en 5-0 sejr over Esbjerg, der dog stadig har svært ved at score i de seneste to kampe i Superligaen. *Esbjerg put Viborg in their place with a 5-0 victory over Esbjerg, who still have difficulty scoring in the last two games in the Super League.*
DanSumT5$_{small}$	Esbjerg satte Viborg på plads i Superligaen efter en skuffende 7.-plads. Viborg er ude af sæsonen med en 6-0-sejr på 6-0 over Viborg. - Sport *Esbjerg put Viborg in place in the Super League after a disappointing 7th place. Viborg is out of the season with a 6-0 win of 6-0 over Viborg. - Sports*

Table 4

Origin	Summary
Reference	Superstjernen havde sine hunde med til Australien på ulovlig vis – det kan komme til at koste dyrt *The superstar brought his dogs to Australia illegally – it could be expensive*
DanSumT5$_{large}$	Hollywood-stjernen risikerer fængselsstraf for at have smuglet sine to hunde til Australien, hvor han filmede med "Pirates of the Caribbean"-filmene. • Johnny Depp og Amber Heard sender hundene hjem • Billedserie: Johnny Deeps venner i Australien • Depp: Jeg elsker dem • Politiker: De er slemme at smide os ud • Politi: De elsker hundene • Politiet: De har svært ved at slippe for straf • Depps hunde er farlige for *The Hollywood star faces jail time for smuggling his two dogs to Australia, where he filmed the "Pirates of the Caribbean" movies. • Johnny Depp and Amber Heard send the dogs home • Photo series: Johnny Depp's friends in Australia • Depp: I love them • Politician: They are mean to throw us out • Police: They love the dogs • Police: They have a hard time escaping punishment • Depp's dogs are dangerous to*
DanSumT5$_{base}$	Johnny Depp og hans kone Amber Heards ulovlige medbringelse af deres hunde til Australien kan koste dem op til ti års fængsel. - Kendte *Johnny Depp and his wife Amber Heard's illegal bringing of their dogs to Australia could cost them up to ten years in prison. - Famous people*
DanSumT5$_{small}$	Johnny Depps ulovlige medbringning af deres to hunde i Australien må måske gå i fængsel for deres ulovlig medbringing. - Film og TV *Johnny Depp's illegal bringing of their two dogs in Australia may have to go to jail for their illegal bringing. - Movies and TV*

Table 5

Origin	Summary
Reference	Hvad får voksne mennesker over 40 til at bruge flere timer om ugen på at vække eventyret til live i den virtuelle verden? De gængse svar på spørgsmålet bunder i fordomme, mener to inkarnerede computerspillere. Her følger deres egen forklaring fra det virtuelle univers *What makes adults over 40 spend several hours a week bringing the adventure to life in the virtual world? The usual answers to the question are rooted in prejudice, say two inveterate computer gamers. Here follows their own explanation from the virtual universe*
DanSumT5$_{large}$	Interview: Det er svært at forestille sig, at livet uden for murene i Danmark kan blive bedre, end man umiddelbart skulle tro. Det kræver i hvert fald en særlig indsats, siger forfatteren Mathiador, der er aktuel med bogen "Kroniken" om en ung mands liv i skoven bagved ved Grønland. Han fortæller, hvordan det er at blive klogere på livet ude i verden, og hvorfor det er sådan en *Interview: It is hard to imagine that life outside the walls of Denmark can be better than one would initially think. In any case, it requires a special effort, says the author Mathiador, who is current with the book "The Chronicle" about a young man's life in the forest behind in Greenland. He tells what it is like to learn more about life out in the world, and why it is such a*
DanSumT5$_{base}$	Interview: Journalisten Mathiador er en af de danske forfattere, der har skrevet bogen "journalisten", der handler om livet som journalist. Han er aktuel med en ny bog om bogen, der er fyldt med højdepunkter, fortæller han i denne uges udgave af'Journalisten'. I dag er han aktuel i en bog, som er baseret på bogen 'Avisen', som handler direkte om journalisten og skuespilleren, der skriver klummer i Lokalavisen Assens. *Interview: The journalist Mathiador is one of the Danish authors who wrote the book "the journalist", which is about life as a journalist. He is current with a new book about the book, which is full of highlights, he tells in this week's edition of 'Journalisten'. Today he is current in a book which is based on the book 'Avisen', which is directly about the journalist and actor who writes columns in the local newspaper Assens.*
DanSumT5$_{small}$	Den danske forfatter og forfatter Mathiador fortæller om sin evne til at blive kendt for sin kærlighed til livet i Danmark. Han er kendt som " Journalisten", men i dag er han aktuel med at gøre det svært for ham at komme i gang med at finde en plads i rummet, hvor han er i centrum af en række af verdens mest prestigefyldte kunstnere, der elsker at skabe kærligheden. Det er en sjælden kærlighed, når han rejser sig til landet. *The Danish writer and author Mathiador talks about his ability to become known for his love of life in Denmark. He is known as "The Journalist", but today he is current with making it difficult for him to start finding a place in the room where he is at the center of a number of the world's most prestigious artists who love to create love. It is a rare love when he stands up to the country.*

Table 6

Origin	Summary
Reference	De danske vikinger leverede en fejlfri præstation under den første prøve ved Eurovision. Men ikke alle elementer fungerede optimalt. *The Danish Vikings delivered a flawless performance during the first rehearsal at Eurovision. But not all elements worked optimally.*
DanSumT5$_{large}$	Det danske Grand Prix-hold havde problemer med sneen til prøverne tirsdag eftermiddag. *The Danish Grand Prix team had problems with the snow for the rehearsals Tuesday afternoon.*
DanSumT5$_{base}$	Rasmussen og hans vikinger havde tirsdag første prøve i Eurovision Song Contest, men sneen var det største problem. Se videoen her på siden og læs mere. *Rasmussen and his Vikings had their first rehearsal in the Eurovision Song Contest on Tuesday, but the snow was the biggest problem. Watch the video on this page and read more.*
DanSumT5$_{small}$	Det danske Grand Prix-håb Rasmussen blev tirsdag eftermiddag ramt af en snestorm, der ifølge DRs underholdningschef blev en rigtig god oplevelse for de danske deltagere i Melodi grand Prix i Portugal. *The Danish Grand Prix hope Rasmussen was hit by a snowstorm on Tuesday afternoon, which according to DR's entertainment manager was a really good experience for the Danish participants in the Melodi Grand Prix in Portugal.*

Table 7

Origin	Summary
Reference	Flere anmeldere roser filmen om forhistorien til Harry Potter for at være underholdende og velsmurt. *Several reviewers praise the film about the prequel to Harry Potter for being entertaining and well-oiled.*
DanSumT5$_{large}$	J. K. Rowlings Potter-filmatisering får ros af anmelderne, der roser instruktøren, men savner dramatik og en knap så nuttet hovedrolleindehaver. *J. K. Rowling's Potter film adaptation is praised by critics, who praise the director but missing drama and a not-so-cute lead.*
DanSumT5$_{base}$	Harry Potter-fans kan godt begynde at glæde sig, hvis de følger instruktøren David Yates' særlige præstation i 'Fantastiske skabninger og hvor de findes', mener flere anmeldere - og det er svært at se filmen, siger de. *Harry Potter fans can start to rejoice if they follow the director David Yates' special performance in 'Fantastic Beasts and Where to Find Them', several reviewers believe - and it is difficult to watch the film, they say.*
DanSumT5$_{small}$	J.K. Rowlings manuskript til Harry Potter-serien er særligt hendes præstation, men filmen er i fuld gang med at underholde publikum. *J. K. Rowling's screenplay for the Harry Potter series is especially her achievement, but the film is busy entertaining audiences.*

Table 8

CaptainA - A mobile app for practising Finnish pronunciation

Nhan Phan
Aalto University
nhan.phan@aalto.fi

Tamas Grosz
Aalto University
tamas.grosz@aalto.fi

Mikko Kurimo
Aalto University
mikko.kurimo@aalto.fi

Abstract

Learning a new language is often difficult, especially practising it independently. The main issue with self-study is the absence of accurate feedback from a teacher, which would enable students to learn unfamiliar languages. In recent years, with advances in Artificial Intelligence and Automatic Speech Recognition, it has become possible to build applications that can provide valuable feedback on the users' pronunciation. In this paper, we introduce the CaptainA app explicitly developed to aid students in practising their Finnish pronunciation on handheld devices. Our app is a valuable resource for immigrants who are busy with school or work, and it helps them integrate faster into society. Furthermore, by providing this service for L2 speakers and collecting their data, we can continuously improve our system and provide better aid in the future.

1 Introduction

Proper pronunciation is needed to build confidence in second language (L2) learners and is essential for effective communication and language acquisition (Gilakjani, 2012). L2 adult learners, who might not have regular exposure to the target language during their everyday life, may lack sufficient opportunities to practise and receive corrective feedback.

With recent advances in Automatic Speech Recognition (ASR) technologies, computer-assisted pronunciation training (CAPT) apps have become more and more effective in helping L2 learners. These apps can immediately give the users feedback on their pronunciation at their convenience. However, while popular languages such as English have many pronunciation applications (Kholis, 2021; Fouz-González, 2020; Wellocution, 2023), there are fewer resources available for Finnish L2 learners. To the best of our knowledge, there was no similar app for CAPT in Finnish before this work.

The main challenge in developing CAPT applications for Finnish and other low-resource languages is the lack of data from L2 speakers. Furthermore, if the L2 corpus is not annotated at the phoneme level, it makes developing an app for mispronunciation detection (MD) more complicated. We designed our CaptainA app to function as well as possible using all available data and add the possibility of collecting users' data after the pilot phase (figure 1). Such information will help evaluate the app's effectiveness for language training and improve our model's performance to better address students' needs in later versions.

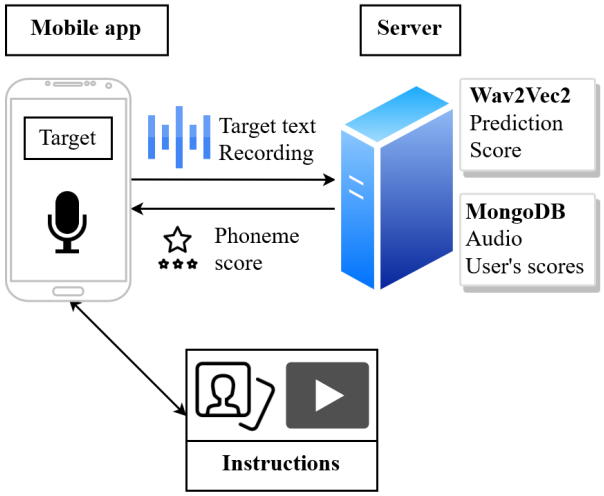


Figure 1: CaptainA app processing flowchart

Recent works from Wu et al. (2021) and Xu et al. (2021) have demonstrated the effectiveness of end-to-end systems with Transformer-based architectures for English MD. While we focus more on practicality, we use a similar approach without a detailed annotation dataset for Finnish.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 265–270

2 Dataset

One of the major challenges that we needed to overcome was the limited data at our disposal. We should note that for the English language, several datasets are available with phoneme level annotation (Zhao et al., 2018; Zhang et al., 2021; Weinberger, 2015). Unfortunately, no such public Finnish resources exist. Thus we opted to use the data collected during the Digitala project (Al-Ghezi et al., 2023) as our primary corpus. This dataset includes ratings from language experts on pronunciation, fluency, lexical, grammatical and the holistic overall level for each audio file, but it does not have phoneme level information.

The Digitala corpus consists of free-form and read-aloud speech, from which we selected 768 short read-aloud samples as those matched our intended scenario most closely. This gave us approximately 60 minutes of audio with the overall pronunciation ratings ranging from 1 to 4, with 4 being the best. The rating is for the whole pronunciation task and not individual phonemes. The lowest pronunciation level (1) contains approximately 2,200 phones, the highest one (4) has only 576 phones, while the remaining 14,000 phones are split almost equally between levels 2 and 3. The corpus was also transcribed by third parties who were not language experts.

The small size of the Digitala corpus and the lack of phoneme annotation meant it was not suitable for training or finetuning for the MD task. However, as there were no better alternatives, we used the Digitala read-aloud transcript as a replacement for the evaluation set. Consequently, we needed another dataset to train our models. After some preliminary experiments, we selected the Finnish Parliament corpus (Kielipankki, 2022), a publicly available corpus without any statistically significant use of dialects (Virkkunen et al., 2023). By training our models for the ASR task with suitably chosen native speakers' samples, we expected the models could learn the features of native Finnish speech and have the potential to identify deviations made by L2 speakers. As a first step, we filtered the most suitable portion of the data, by selecting speeches with low or average speaking rates (which is the most similar to how L2 learners speak). As an additional step, we also restricted the data by excluding older (50+) speakers, since our target audience is generally younger immigrants. The last step in data preparation was the splitting of the 281 hours of data into 75% for training, and 25% for tuning hyperparameters and evaluating the speech recognition models. We should note that we also used two publicly available reference models, called Finnish-NLP[1] and Finnish-NLP-S [2]. Both have been trained with 228 hours of Finnish Parliament data and approximately 47 hours of data from other sources.

3 Implementation

3.1 Server

The core technology inside our server is based on wav2vec 2.0 (Baevski et al., 2020), which was already proven to work exceptionally well even with very limited amount of data (Wu et al., 2021; Xu et al., 2021). We selected XLS-R (Babu et al., 2022) and Uralic, a subset of VoxPopuli (Wang et al., 2021), as our pre-train models, and use the state-of-the-art model in Finnsh ASR, Finnish-NLP, as our baseline. Except for entropy β, all models used the same hyperparameters, and there is no language model used for decoding.

Leveraging the phonetic nature of the Finnish language, where each phoneme is represented by exactly one grapheme[3], we can use graphemes as output units during the ASR training procedure. Once the ASR models were trained, we used the forced alignment algorithm for Connectionist Temporal Classification (CTC) from Kürzinger et al. (2020) to determine the success of pronunciation. This algorithm provides both time alignment and a probability score for each grapheme. Inspired by the traditional Goodness of Pronunciation method (Witt and Young, 2000), we use such information to generate feedback for the user.

One major issue we had to overcome was the overconfidence of the wav2vec 2.0 models. As it is well known, the CTC algorithm often results in spiky outputs (Zeyer et al., 2021), which in terms would mean that we can only provide binary (correct/incorrect) feedback to the user. Naturally, a good pronunciation training app should give more detailed information (Engwall and Bälter, 2007), thus, reducing the peakedness of the outputs was important. To achieve this, we chose the negative maximum entropy regularization technique Liu

[1] https://huggingface.co/Finnish-NLP/wav2vec2-xlsr-1b-finnish-lm-v2

[2] https://huggingface.co/Finnish-NLP/wav2vec2-xlsr-300m-finnish-lm

[3] except "nk" [ŋk] and "ng" [ŋː]

Model	Vocabulary	Parameters	Entropy β	CER	Recall	Precision	$\mathbf{F_1}$
Finnish-NLP	Grapheme	1bil	0%	**15.4%**	59.8%	**33.3%**	**42.8%**
Finnish-NLP-S		300mil		22.3%	**65.0%**	26.1%	37.2%
XLS-R	Grapheme	300mil	0%	20.9%	61.1%	26.7%	37.2%
XLS-R-5	Grapheme		5%	**19.5%**	63.1%	**30.0%**	**40.6%**
XLS-R-10	Grapheme		10%	21.2%	63.1%	29.4%	40.1%
XLS-R-10-P	Phoneme		10%	21.3%	63.2%	27.3%	38.1%
Uralic-10	Grapheme		10%	30.4%	64.3%	23.4%	34.3%
Uralic-10-P	Phoneme		10%	29.6%	**66.8%**	22.6%	33.8%

Table 1: Speech models' performance in ASR and MD on Digitala read-aloud set.

et al. (2018) during training, which redistributes $\beta\%$ of the total probability mass uniformly to all outputs, ensuring the smoothness of the final predictions.

3.2 Mobile app

We use Unity (Juliani et al., 2020) as our development engine. With Unity we can simultaneously publish our CaptainA app to multiple platforms: Android, iOS and Windows. Our app contains various study materials, and Unity Editor allows us to easily integrate those multimedia content into the app. We make use of the engine to visualize our pronunciation instructions with animations and limit the rest to simple UI, thus lowering the application's power consumption.

Arapakis et al. (2021) estimated a 7 seconds threshold where mobile (web search) users' experience decreases significantly. To maintain a reasonable response time, we use a manual VAD system to remove the silent parts from the recording: the users must press and hold the record button to record their audio samples.

The app supports two modes; the "Topic" mode supplies curated words and phrases for various topics (Easy, Normal, Hard, Greetings, Grocery, similar vowels pair, or classic Finnish literature...), often along with English translation and audio samples from native speakers. On the other hand, the "Freestyle" mode enables users to practice any word or phrase by first prompting for the text that the user will attempt to pronounce.

The score for each phoneme is saved locally, enabling users to track their progress. The data is valuable in developing speech applications for L2 speakers. In the future, with the users' permission, we can collect their records to evaluate the app's effectiveness and other metadata.

CaptainA also provides pronunciation instructions via sample audios, pictures, animations and videos, which are beneficial for users during self-practice (Engwall and Bälter, 2007). The audio, photo and animation materials are directly stored in the app, while the videos are accessible via a public, ad-free platform. We should note that external links would generally have an adverse effect on user experience, still we choose this solution to supply high-quality tutorial videos while keeping the size of the app reasonably small.

4 Results

To validate our models, we computed their character error rate (CER), Recall (percentage of mispronunciations correctly detected) and Precision (the ratio of detected mispronunciations actually being mispronunciation, according to a native Finnish listener) using the Digitala read-aloud corpus. The empirical results can be seen in Table 1. The first thing that we noticed is that the large Finnish-NLP produced significantly lower and the small Finnish-NLP-S higher CER compared to the majority of our models. Next, we compared the models in terms of MD and saw that Finnish-NLP yielded the highest overall F_1 score. However, the smaller XLS-R-5 and XLS-R-10 managed to achieve comparable results with the help of entropy regularization.

The benefit of entropy regularization is seen when we increase the value of β and note that both Recall and Precision also increase. From our experiment, we found that β between 5% and 10% produces the best result for MD task. Looking at the detailed breakdown in table 2, we also found that, the smaller XLS-R outperformed the Finnish-NLP in Recall for pronunciation level 1 samples, while slightly falling behind in Precision. The gap in Precision widens as the speakers' pronunciation skill improves. Considering the practicality of

Model	CER	Recall	Precision
Finnish-NLP	**26.9%**	72.6%	**38.7%**
XLS-R-5	31.4%	77.4%	36.2%
XLS-R-10	33.5%	**78.5%**	36.8%
Finnish-NLP	**20.0%**	61.5%	**32.7%**
XLS-R-5	24.1%	**63.3%**	29.1%
XLS-R-10	24.7%	63.2%	28.9%
Finnish-NLP	**11.6%**	42.4%	**27.2%**
XLS-R-5	15.4%	**46.7%**	24.1%
XLS-R-10	17.6%	45.7%	22.2%
Finnish-NLP	**6.0%**	18.8%	**20.0%**
XLS-R-5	10.3%	**25.0%**	16.0%
XLS-R-10	13.6%	**25.0%**	10.0%

Table 2: CER, Recall and Precision for the pronunciation levels 1 to 4 (top to bottom: worst to best)

smaller models, they would be suitable in MD for beginner L2 learners. While the Uralic model did, in our preliminary experiment on Common Voice 7.0 test set, produce lower CER on native Finnish speakers, it failed in both ASR and MD task on L2 speakers. One possible reason is that the Uralic models were not exposed to foreign language families, unlike the XLS-R models.

One common type of mistake made by Finnish L2 speakers is related to the duration of phonemes. In our test set, excluding other types of mistakes, the XLS-R-10 model achieved approximately 68% Recall rate where users pronounce short (single) phonemes too long, and 57% where users pronounce long (double) phonemes too short.

Although our goal was to develop a CAPT app that works well for all users, due to the composition of the training data and the randomness in the optimization process, the system still developed some biases towards subgroups of users. For samples in the same pronunciation level, the XLS-R-10 model has better MD for *male* speakers, with a higher Recall rate (66.0% vs *females* 59.8%), and a slightly higher Precision (31.2% vs 27.6%). The gender distribution in the test set is balanced, with 53% *male* samples and 47% *female*. We will release more detailed analyses in the future once we collect more data from users.

While it is possible to use the training part of the Digitala corpus for finetuning our wav2vec 2.0 models, we could not control the pronunciation quality, as the speakers are L2 learners and there is no phoneme annotation. In our preliminary experiments we found that finetuning with bad pronunciation data led to lower performance in MD.

5 Self-study assistant

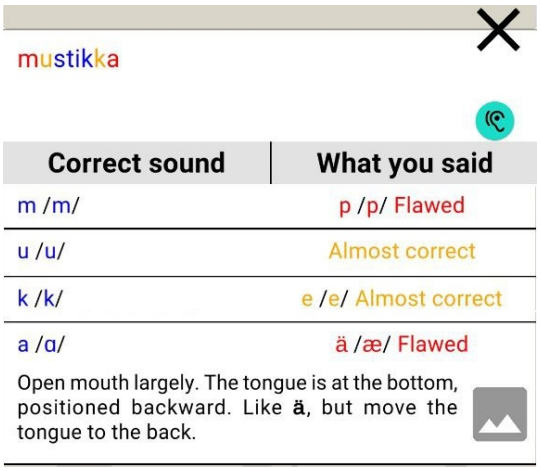


Figure 2: The result is coloured based on pronunciation score.

CaptainA (see figure 1) allows users to enter words into a text prompt to practise pronunciation. Their audio is sent to the server, and the device will display the obtained rating for each phoneme, with three possible ratings in colors (figure 2): flawed (phoneme is not recognizable), almost correct (improved, but not clear), and correct. The "almost correct" rating is given as positive feedback when user's phoneme score improves, but is still not considered correct, as suggested by Engwall and Bälter (2007).

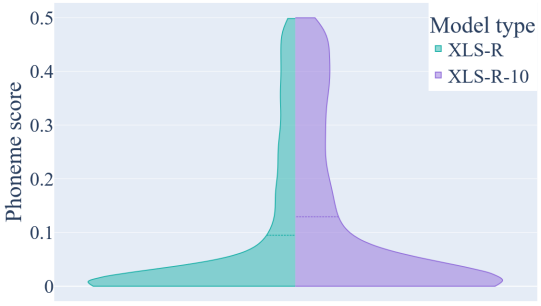


Figure 3: Phoneme score distribution for phonemes with scores less than 0.5 in the test set.

We based our ratings on the phoneme score distribution of the test set. With model XLS-R-10, any mispronounced phoneme with a score higher than 0.2 (higher than 70% percentile of

all mispronounced phonemes), or correct pronounced phoneme with a score lower than 0.8 (lower than 10% percentile of all correct pronounced phonemes) is classified as the intermediate rating "almost correct". In CAPT, we consider the cost of misclassifying correct pronunciation as incorrect more severe (Bachman et al., 1990), and therefore mispronunciation has a higher chance of being classified to a higher rating. It should be noted, we were able to introduce an additional rating category from the binary incorrect/correct dataset by implementing the entropy regularization described earlier. Higher entropy is also the practical reason to select the XLS-R-10 model instead of the XLS-R-5 for CaptainA.

When making mispronunciation, the users are also advised to refer to the app multimedia pronunciation instructions (figure 4).

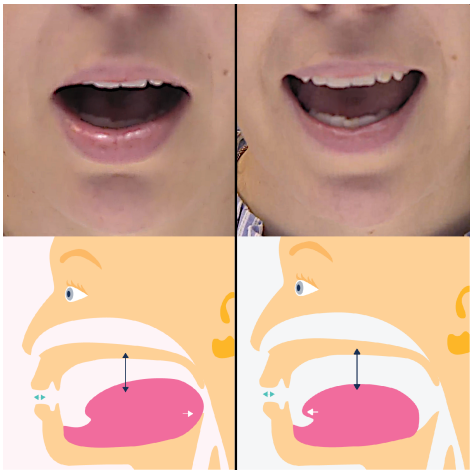


Figure 4: Visual pronunciation instructions for A [ɑ] (left) and Ä [æ] (right).

6 Conclusion

In this paper, we presented the prototype of CaptainA, an app that helps language learners practise Finnish pronunciation. Because of the lack of data available for phoneme level pronunciation mistakes, our solution is based on multilingual wav2vec 2.0 models, which are finetuned for native Finnish ASR. By running the L2 learners' utterances through the ASR without a language model, we predict pronunciation errors and probability scores that indicate the success of pronunciation. The resulting models are validated by measuring CER, Recall and Precision for samples of different levels of pronunciation judged by human experts. In the future, we plan to collect user data (feedback and audio) with our app to update the models and improve the self-study application.

CaptainA was trained on a native speaker corpus and evaluated on 1 hour of transcribed L2 data, without requiring detailed annotation from experts or fine-tuning L2 data. Its performance could be a motivation to apply a similar setup to other low-resource languages. Based on the initial result of CaptainA, we plan to develop a similar CAPT app for Finland Swedish.

Acknowledgments

The computational resources were provided by the Aalto Science-IT. CaptainA was funded by the Kielibuusti project. This work was also supported by NordForsk through the funding to Technology-enhanced foreign and second-language learning of Nordic languages, project number 103893. The video and photo instructions were made with the help of the Aalto University Language Centre. The side mouth illustrations are from Aino Huhtaniemi.

References

Ragheb Al-Ghezi, Yaroslav Getman, Ekaterina Voskoboinik, Mittul Singh, and Mikko Kurimo. 2023. Automatic rating of spontaneous speech for low-resource languages. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 339–345. IEEE.

Ioannis Arapakis, Souneil Park, and Martin Pielot. 2021. Impact of response latency on user behaviour in mobile web search. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, pages 279–283.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2022. https://doi.org/10.21437/Interspeech.2022-143 XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale. In *Proc. Interspeech 2022*, pages 2278–2282.

Lyle F Bachman et al. 1990. *Fundamental considerations in language testing*. Oxford university press.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Olov Engwall and Olle Bälter. 2007. Pronunciation feedback from real and virtual language teachers.

Computer Assisted Language Learning, 20(3):235–262.

Jonas Fouz-González. 2020. Using apps for pronunciation training: An empirical evaluation of the english file pronunciation app. *Language, Learning and Technology*, 24.

Abbas Pourhosein Gilakjani. 2012. A study of factors affecting efl learners' english pronunciation learning and the strategies for instruction. *International journal of humanities and social science*, 2(3):119–128.

Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. 2020. http://arxiv.org/abs/1809.02627 Unity: A general platform for intelligent agents.

Adhan Kholis. 2021. Elsa speak app: automatic speech recognition (asr) for supplementing english pronunciation skills. *Pedagogy: Journal of English Language Teaching*, 9(1):01–14.

Kielipankki. 2022. http://urn.fi/urn:nbn:fi:lb-2022052002 Aalto Finnish Parliament ASR Corpus 2008-2020, version 2.

Ludwig Kürzinger, Dominik Winkelbauer, Lujun Li, Tobias Watzel, and Gerhard Rigoll. 2020. CTC-segmentation of large corpora for german end-to-end speech recognition. In *International Conference on Speech and Computer*, pages 267–278. Springer.

Hu Liu, Sheng Jin, and Changshui Zhang. 2018. Connectionist temporal classification with maximum entropy regularization. *Advances in Neural Information Processing Systems*, 31.

Anja Virkkunen, Aku Rouhe, Nhan Phan, and Mikko Kurimo. 2023. https://doi.org/10.1007/s10579-023-09650-7 Finnish parliament ASR corpus: Analysis, benchmarks and statistics. *Language Resources and Evaluation*, pages 1–26.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. https://doi.org/10.18653/v1/2021.acl-long.80 VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 993–1003, Online. Association for Computational Linguistics.

Steven Weinberger. 2015. http://accent.gmu.edu *Speech Accent Archive*. George Mason University.

Wellocution. 2023. https://www.boldvoice.com/ Boldvoice web page.

Silke M Witt and Steve J Young. 2000. Phone-level pronunciation scoring and assessment for interactive language learning. *Speech communication*, 30(2-3):95–108.

Minglin Wu, Kun Li, Wai-Kim Leung, and Helen Meng. 2021. Transformer based end-to-end mispronunciation detection and diagnosis. In *Interspeech*, pages 3954–3958.

Xiaoshuo Xu, Yueteng Kang, Songjun Cao, Binghuai Lin, and Long Ma. 2021. Explore wav2vec 2.0 for mispronunciation detection. In *Interspeech*, pages 4428–4432.

Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2021. http://arxiv.org/abs/2105.14849 Why does CTC result in peaky behavior?

Junbo Zhang, Zhiwen Zhang, Yongqing Wang, Zhiyong Yan, Qiong Song, Yu-Kai Huang, Ke Li, Daniel Povey, and Yujun Wang. 2021. https://doi.org/10.21437/interspeech.2021-1259 speechocean762: An Open-Source Non-Native English Speech Corpus for Pronunciation Assessment. *Conference of the International Speech Communication Association*.

Guanlong Zhao, Sinem Sonsaat, Alif Silpachai, Ivana Lucic, Evgeny Chukharev-Hudilainen, John Levis, and Ricardo Gutierrez-Osuna. 2018. L2-arctic: A non-native english speech corpus. In *Interspeech*, pages 2783–2787.

DanTok: Domain Beats Language for Danish Social Media POS Tagging

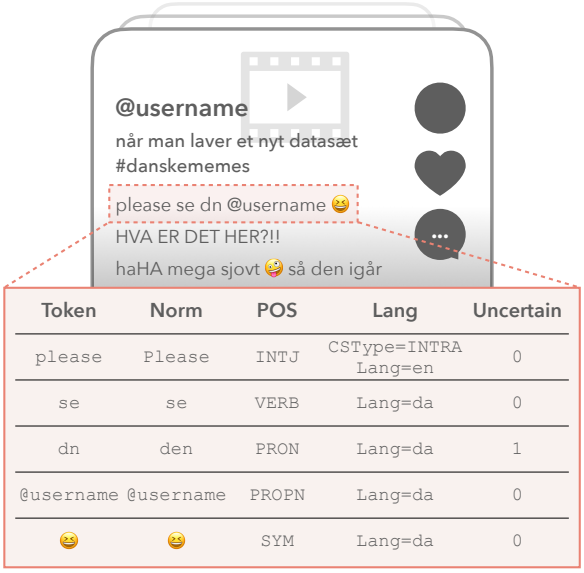
Kia Kirstein Hansen, Maria Barrett, Max Müller-Eberstein, Cathrine Damgaard, Trine Naja Eriksen, Rob van der Goot
IT University of Copenhagen
[kiah, mbarrett, mamy, catd, trer, robv]@itu.dk

Abstract

Language from social media remains challenging to process automatically, especially for non-English languages. In this work, we introduce the first linguistically annotated dataset for TikTok comments and the first Danish social media dataset with part-of-speech annotation. Additionally, we supply annotations for normalization, code-switching, and annotator uncertainty. As transferring models to such a highly specialized domain is non-trivial, we conduct an extensive study into which source data and modeling decisions most impact the performance. Surprisingly, transferring from in-domain data, even from a different language, outperforms in-language, out-of-domain training. These benefits nonetheless rely on the underlying language models having been at least partially pre-trained on data from the target language. Using our additional annotation layers, we analyze how normalization, code-switching, and human uncertainty affect the tagging accuracy.

1 Introduction

Language data from social media offer unique insights into how communities use language to communicate in a natural, spontaneous setting, using a highly domain-specific vocabulary. This domain is, however, also subject to frequent changes, high noise, and variability, making it difficult to process (Eisenstein, 2013) both for high (Gimpel et al., 2011; Derczynski et al., 2013) and especially for lower-resourced languages (Kaji and Kitsuregawa, 2014; Albogamy and Ramasy, 2015; Singh et al., 2018; Mæhlum et al., 2022). To better understand and improve how to target such a highly specialized domain, we present the first linguistically annotated dataset of contemporary Danish social media from TikTok, a relatively new platform focused on short videos. The Danish Dependency Treebank (Kromann et al., 2003) from the PAROLE Corpus (Bilgram and Keson, 1998), and its Universal Dependencies (UD) conversion (Johannsen et al., 2015) is the primary part-of-speech (POS) resource for Danish (Kirkedal et al., 2019). However, it contains well-edited text only.



Token	Norm	POS	Lang	Uncertain
please	Please	INTJ	CSType=INTRA Lang=en	0
se	se	VERB	Lang=da	0
dn	den	PRON	Lang=da	1
@username	@username	PROPN	Lang=da	0
😆	😆	SYM	Lang=da	0

Figure 1: DanTok annotation layers for a fabricated example comment.

We present not only the first Danish social media dataset with POS annotation, but, to the best of our knowledge, also the first linguistic exploration of TikTok language in any language. The dataset contains annotation for POS, normalization, code-switching and annotator uncertainty (Figure 1). It further prioritizes quality over quantity, covering over 8k tokens, which have been manually verified to ensure relevance, correctness, and privacy.

It has been shown that training language models on the target domain can be used for cross-domain

Code available at: github.com/kkirsteinhansen/dantok. Data is available upon request (contact robv@itu.dk).

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 271–279

learning (Gururangan et al., 2020; Barbieri et al., 2022). In this work, we aim to tease apart the effect of the pre-training data of the language model and the source of the annotated training data for the target task. For this purpose, we contribute: ① DanTok, the first linguistically annotated dataset of TikTok comments and the first Danish POS-annotated social media dataset; ② An extensive study of transfer learning targeting this highly specialized domain; ③ An in-depth analysis investigating the unique features of this new domain, and how specific data/model properties help boost transfer performance.

2 DanTok

2.1 Collection

The social media platform TikTok shows a continuous, personalized feed of short videos on various topics. Users may respond to these videos using likes and/or comments. In 2022, the number of Danish users of age ≥ 18 was estimated to be 1.14M. This is substantial compared to the total population in Denmark (5.8M), as well as, e.g., the number of Danish Twitter users (est. 685k).[1] It is challenging to gather data from TikTok in a reproducible way due to algorithmically populated user feeds as well as the inability to filter videos by language. To increase reproducibility, we browsed videos directly by hashtag while logged out. We selected 15 Danish hashtags (Appendix B.1) where videos were determined to be predominantly in Danish.

Filtering and Deduplication We applied automatic filtering to maximize data diversity relative to the dataset size while reducing annotation workload. Using the Aspell dictionary for Danish (Aspell, 2019), we removed non-Danish comments while retaining interesting token variations using a removal threshold of $< 10\%$ or $> 60\%$ OOV (out-of-vocabulary) tokens per comment. As comments often have a high degree of repetitiveness (e.g., "wow", "woow!!"), we deduplicated them by iteratively merging the two most similar comments, keeping only one, until there were no more pairs with a similarity greater than a threshold t. Similarity was measured using the normalized Levenshtein distance, and we retained the comment with a higher total similarity to all other comments (i.e., the more prototypical one). To determine the best threshold, we conducted a sweep over $t \in (0.0; 1.0)$ in 0.05 increments (see also Appendix B.2), and chose $t = 0.60$, as this set retained a relatively high diversity while accounting for the downstream, manual filtering of irrelevant comments down to our target of annotating 8k tokens.

2.2 Annotation

The data was tokenized using the `NLTK TweetTokenizer` (Bird et al., 2009) with manual post-correction. We replaced usernames with *@username*, and manually filtered out comments containing personal information. After two rounds of annotation[2], we reached a Fleiss' κ score of 96.05. As the remaining disagreements were mainly ambiguities and accidental mistakes, we annotated the remainder of the dataset once, with the following layers:

POS We followed the UD 2.11 guidelines and the social media-specific suggestions by Sanguinetti et al. (2020). Where possible, we annotated the intended meaning of a token given its use in context, e.g., in "Would you like a 🍺?", 🍺 is a `NOUN`. Tokens that imitate pronunciation were tagged as `INTJ`. *Den*, *det* and *de*, which may function as articles or pronouns, were tagged as `DET` only when followed by an adjective.

Normalization Given that we annotated the intended meaning of a token, we additionally decided to annotate for lexical normalization. We followed the format and guidelines of MultiLexNorm (van der Goot et al., 2021a): URLs and interjections were not normalized, and splits and merges were included in the annotations. Merging of tokens that had erroneously been written as multiple words is indicated in the normalization column of the last word, as compound words in Danish adopt the part-of-speech (POS) of the final word. For tokens that were split, an overall POS tag was given to this token alongside individual tags for the tokens resulting from the splitting.

Code-Switching Code-switching is indicated with language and type. The type is either `INTER` (full sentence), `INTRA` (individual tokens) or `MIXED` (a lemma from one language with inflection from another language) as described by

[1] https://datareportal.com/reports/digital-2022-denmark

[2] See Appendix A for information about the annotators.

	Langs	Train-sources	Type	Train-size	Architecture	Params	%Unk.	Subw. ratio	Reference
LLM									
DANISH-BERT-botxo	DA	web, wiki, subtitles		10gb	Bert-base	111M	0.15	1.28	github.com/certainlyio/nordic_bert
RØBÆRTA-base-danish	DA	web	-D+L	?	Roberta-base	125M	0.00	1.58	hf.co/DDSC/roberta-base-danish
ÆLÆCTRA-danish-small-cased	DA	legal, social, web, wiki, news		1,045M words	Electra-small	14M	0.04	1.39	github.com/MalteHB/-l-ctra
BERTWEET-Base	EN	social		850M tweets	Roberta-base	135M	0.10	1.65	Nguyen et al. (2020)
BERTWEET-Large	EN	social	+D-L	850M tweets	Roberta-large	355M	0.00	1.90	Nguyen et al. (2020)
TWITTER-ROBerta-base	EN	social		58M tweets	Roberta-base	125M	0.00	1.90	Barbieri et al. (2020)
TWITTER-XLM-roberta-base	30+	social		198M tweets	XLM-r base	278M	0.01	1.45	Barbieri et al. (2022)
BERNICE	66	social	+D-L	2.5B tweets	Roberta-base	278M	0.00	1.44	DeLucia et al. (2022)
TWHIN-bert-large	100+	social		7.5B tweets	new	561M	0.01	1.45	Zhang et al. (2022)
TREEBANK									
LINES	EN	fiction, nonfiction, spoken	-D-L	57,372 words					Ahrenberg (2015)
TWEEBANK2	EN	social	+D-L	24,753 words					Liu et al. (2018)
DDT	DA	fiction, nonfiction, spoken, news	-D+L	80,378 words					Johannsen et al. (2015)

Table 1: An overview of the used language models and POS fine-tuning sets. %Unk. is the percentage of unknown subwords in our development data; Subw. ratio is the average amount of subwords per word. Capitalized name parts are handles.

	COMMENTS	TOKENS	TYPES	TTR
Dev	429	4,000	1,520	0.38
Test	430	4,028	1,519	0.38
Total	859	8,028	2,512	0.31

Table 2: DanTok dataset statistics. TTR is type-token ratio.

Sanguinetti et al. (2020). We used the Danish dictionary[3] for cross-referencing, as many originally English words are now considered Danish.

Certainty Following Bassignana and Plank (2022), the annotator's certainty of a POS tag was annotated as either 0 (certain) or 1 (uncertain).

2.3 DanTok Statistics

Our final dataset consists of 8,028 tokens and 2,512 unique types (Table 2). A comparative POS tag distribution is given in Appendix C. In DanTok, we observe that 16.66% of the tokens required normalization, 5.03% were code-switched (all to English), and 5.12% had annotation uncertainty. Overall, these annotation layers allow us to investigate how Danish is used on contemporary internet platforms with respect to syntax, and how sociolinguistic factors such as code-switching can impact downstream performance.

3 Experiments

3.1 Setup

For a highly specialized dataset such as DanTok, transfer learning is key, as there is no training data matching the domain and language. We therefore investigated 36 combinations of in/out-of-domain (+D/-D)[4] and in/out-of-language (+L/-L)[5] training data and large language models (LLMs). We selected English as the -L transfer language due to dataset and language model availability. All experiments were replicated on the normalized version of DanTok. The Danish LLMs are trained on web data, including some forum data, but none are explicitly optimized for social media. The LLMs and training sets used in our experiments are given in Table 1. All the models consist of an LLM encoder plus a linear layer for POS labeling (both fully finetuned) and are implemented in MaChAmp v0.4 (van der Goot et al., 2021b) using default hyperparameters with the development data for model selection. To avoid overfitting on DanTok, we use the transfer data's development set for model selection (Artetxe et al., 2020).

3.2 Results

Our main results are given in Table 3. Unsurprisingly, the combination of in-domain, in-language (+D+L) training data and LLMs results in the best overall performance. In general, having in-language data is more beneficial than in-domain data; however, when training on a single dataset, the in-domain English dataset (+D-L) leads to surprisingly high performance with the multilingual language models, even outperforming all scores obtained with the Danish training data (-D+L). One reason for this could be the relatively high frequency of code-switched tokens (5%). Inter-

[3] https://dsn.dk/ordboeger/retskrivningsordbogen/

[4] +D: social media data, -D: data from other domains.

[5] +L: trained on Danish, -L: trained on other languages.

Model \ Data	-D-L LINES	+D-L TWB	-D+L DDT	+D-L + -D+L TWB+DDT
-D+L DANISH-BERT	44.02	49.60	77.98	84.08
-D+L RØBÆRTA	58.43	60.82	70.17	78.72
-D+L ÆLÆCTRA	49.50	63.30	74.20	84.95
+D-L BERTWEET-B	27.80	38.00	67.90	79.47
+D-L BERTWEET-L	25.92	36.55	67.40	81.50
+D-L TWITTER-ROB	25.02	37.30	64.05	79.40
+D+L TWITTER-XLM	67.58	77.15	72.15	83.28
+D+L BERNICE	70.45	78.22	72.95	83.28
+D+L TWHIN	69.30	81.38	72.65	**85.92**

Table 3: POS tagging accuracy on the DanTok development set using combinations of in/out-of-domain (+D/-D) and in/out-of-language (+L/-L) models and training data, plus a concatenation covering +D and +L.

Model \ Data	-D-L LINES	+D-L TWB	-D+L DDT	+D-L + -D+L TWB+DDT
-D+L DANISH-BERT	42.69	48.12	80.19	85.75
-D+L RØBÆRTA	60.79	63.48	73.80	82.36
-D+L ÆLÆCTRA	50.87	61.69	78.48	88.45
+D-L BERTWEET-B	28.24	38.48	70.88	82.83
+D-L BERTWEET-L	27.46	37.48	70.98	85.35
+D-L TWITTER-ROB	25.75	38.13	68.76	84.19
+D+L TWITTER-XLM	69.85	80.12	75.16	86.26
+D+L BERNICE	72.01	80.27	75.61	85.15
+D+L TWHIN	70.95	83.09	75.33	**88.80**

Table 4: POS tagging accuracy on the normalized DanTok development set using combinations of in/out-of-domain (+D/-D) and in/out-of-language (+L/-L) models and training data.

estingly, model size (see Table 1) is not a good predictor of performance: Although the largest model, TWHIN, obtains the highest score overall, it requires large amounts of pre-training data and a specialized pre-training objective based on rich social engagements (Zhang et al., 2022). Meanwhile, ÆLÆCTRA's performance is very close, despite being 41 times smaller. Given these results, we conclude that the best strategy for obtaining a high-quality tagger would be to use domain-specific models when available (even if multilingual) and use in-domain fine-tuning data even if in another language (+ in-language if available).

Table 4 shows that using normalized data gives a consistent boost of 2–5 % points across all setups, with only a few exceptions. Furthermore, performance varies less compared to the non-normalized data (Table 3).

LLM	-NORM	+NORM
TWHIN	86.05	88.18
ÆLÆCTRA	85.80	88.55

Table 5: Results on the DanTok test set of our two best models trained on TWEEBANK and DDT.

On Test Data TWHIN performs similarly on the development and test data. After normalization, the smaller ÆLÆCTRA model outperforms TWHIN slightly (Table 5).

4 Analysis

4.1 Subword Analysis

The Subword ratio (Table 1) does not show a clear correlation with performance, so we qualitatively evaluate the subword segmentation of the two best-performing models, TWHIN and ÆLÆCTRA. Surprisingly, we find that the multilingual model (TWHIN) seems more capable of interpreting inflection suffixes than the Danish model. It correctly splits morphemes indicating definiteness, plurality, or adverbial status, which the Danish model sometimes fails to do. Examples of this are *batterier* ("batteries") split into *batteri-er* ("batteri-es") and *dårligt* ("badly") split into *dårlig-t* ("bad-ly") only by the multilingual model, whereas the Danish model does not split these tokens at all.

4.2 Stratified Analysis

We explore the accuracy on different subsets of the development set according to our additional annotation layers (Table 6). We observe that the models, perhaps unsurprisingly, struggle more with tokens that were normalized, as well as tokens that annotators were also uncertain of. For code-switched tokens, we observe a large performance drop for the in-language LLM (ÆLÆCTRA) despite fine-tuning on English in-domain data. Surprisingly, the multilingual model, likewise fine-tuned on Danish and English in-domain data, also struggles with code-switched tokens.

4.3 Qualitative Error Analysis

The most frequent tag confusions for the best ÆLÆCTRA model are given in Figure 2. TWHIN follows a similar pattern. Over half of the errors made by each tagger on the original data are shared with the other tagger. Some of the errors

LLM	POS Certainty			Normalized			In fine-tune vocab			Code-switched		
	n	-	+	n	-	+	n	-	+	n	-	+
twhin	203	62.1	87.2	3,338	88.9	70.8	1,234	83.3	87.1	3,808	86.3	78.1
ÆLÆCTRA		58.6	86.4		88.6	66.8		83.1	85.8		86.1	62.5

Table 6: Stratified accuracy on the 4,000-token dev set of the two best models trained on TWEEBANK and DDT. n is the number of tokens in the - category, e.g., 1,234 words were not seen during fine-tuning.

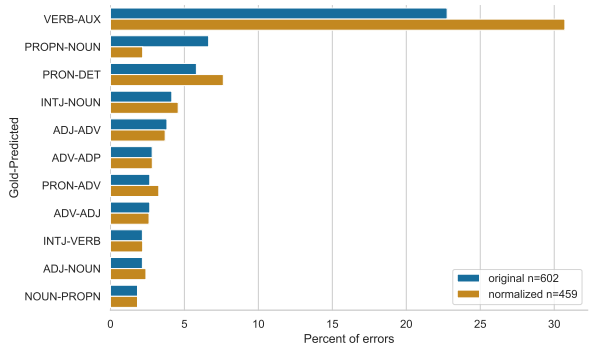


Figure 2: The 11 most frequent tag confusions for the ÆLÆCTRA model.

are caused by erroneous annotations in DanTok. The most frequent error types can be categorized as follows:

VERB vs. AUX In DanTok, the present tense of the copula verb *være* ("to be") has been labeled VERB when it is the only verb in the sentence. However, the models prefer the tag AUX in 91.6% and 85.0% of cases, respectively. This seems to be in line with the UPOS guidelines and is likely a result of the annotation of *er* in the DDT training set; here, 78.85% of *er* tokens have been tagged as AUX (the remaining being tagged as VERB).

Pronoun Confusions Tokens that may be multiple parts of speech confuse the taggers. The most frequent issue is PRON and DET confusion, which is arguably non-trivial in Danish[6]. PRON and ADV confusion is also prevalent; e.g., the token *der* can be either a relative PRON, the preliminary subject "there", or an ADV of place. In the erroneous predictions, *der* is generally tagged as ADV.

Proper Noun Inconsistencies Orthographic variations in social media language throw off the models. For example, names written in lowercase are often tagged as NOUN rather than PROPN. On the normalized data, PROPN (gold) → NOUN errors decreased by 75% for ÆLÆCTRA and 62% for TWHIN. Likewise, when capitalized names are used in context, the models labeled them as PROPN, whereas we annotated the syntactical use of the token, e.g., *filming a TikTok*/NOUN.

ADV vs. ADP These are errors made on tokens like *af* ("of, off") and *for* ("for, too") which may function as both prepositions and adverbs[7]. In a few cases, the models do not recognize when *for* is used as an adverb of degree.

ADJ vs. ADV For adjectives that end in *-t*, the models seem to prefer the ADV tag. While *-t* can indicate an adverb, it may also indicate the gender of an adjective. The token *her* ("here"), an adverb, also poses a challenge when it occurs before a noun, e.g. *den her bog* ("this book"). In such cases, the models seem to prefer the erroneous tag sequence *den*/DET ***her*/ADJ** *bog*/NOUN.

Interjection Confusions Tokens that are meant to imitate pronunciation have been labeled as INTJ in DanTok, but the models seem to prefer a more concrete labeling[8]. The models also prefer INTJ for tokens with character repetition, whereas we tagged these tokens according to their presumed intended function.

5 Conclusion

We presented DanTok, the first linguistically annotated TikTok dataset and the first Danish social media dataset with POS annotation. We conducted an extensive analysis of how to best transfer to a highly specialized domain in a mid-resource language, and we demonstrated that LLMs benefit from common approaches such as normalization, while struggling with the same cases as the human annotators. Simultaneously, our results show that although in-language data and models form the basis for high performance, in-domain data, even from another language, should not be neglected in order to achieve state-of-the-art results.

[6]Consider, e.g., ***den*/PRON** *bog*/NOUN vs. ***den*/DET** *gamle*/ADJ *bog*/NOUN ("**that** book" vs. "**the** old book").

[7]*For* may also be used as a conjunction.

[8]E.g., "It's *nuclear*, not *nucular*," should be tagged as if it said *nuclear* twice.

Acknowledgments

We would like to thank the members of NLPnorth, MaiNLP Lab and the anonymous reviewers for their feedback. This project has been supported by the Pioneer Centre for Artificial Intelligence. Maria Barrett is supported by a research grant (34437) from VILLUM FONDEN. Max Müller-Eberstein is supported by the Danmarks Frie Forskningsfond (DFF) Sapere Aude grant 9063-00077B.

References

Lars Ahrenberg. 2015. Converting an English-Swedish parallel treebank to Universal Dependencies. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 10–19, Uppsala, Sweden. Uppsala University, Uppsala, Sweden.

Fahad Albogamy and Allan Ramasy. 2015. Towards POS tagging for Arabic tweets. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 167–171, Beijing, China. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. 2020. A call for more rigor in unsupervised cross-lingual learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7375–7388, Online. Association for Computational Linguistics.

GNU Aspell. 2019. GNU Aspell. `http://aspell.net/`. Version da-1.6.36-11-0.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 258–266, Marseille, France. European Language Resources Association.

Elisa Bassignana and Barbara Plank. 2022. CrossRE: A cross-domain dataset for relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3592–3604, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Thomas Bilgram and Britt Keson. 1998. The construction of a tagged Danish corpus. In *Proceedings of the 11th Nordic Conference of Computational Linguistics (NODALIDA 1998)*, pages 129–139, Copenhagen, Denmark. Center for Sprogteknologi, University of Copenhagen, Denmark.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Alexandra DeLucia, Shijie Wu, Aaron Mueller, Carlos Aguirre, Philip Resnik, and Mark Dredze. 2022. Bernice: A multilingual pre-trained encoder for Twitter. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6191–6205, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 198–206, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, Georgia. Association for Computational Linguistics.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for Twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 42–47, Portland, Oregon, USA. Association for Computational Linguistics.

Rob van der Goot, Alan Ramponi, Arkaitz Zubiaga, Barbara Plank, Benjamin Muller, Iñaki San Vicente Roncal, Nikola Ljubešić, Özlem Çetinoğlu, Rahmad Mahendra, Talha Çolakoğlu, Timothy Baldwin, Tommaso Caselli, and Wladimir Sidorenko. 2021a. MultiLexNorm: A shared task on multilingual lexical normalization. In *Proceedings of the 2021 EMNLP Workshop W-NUT: The Seventh Workshop on Noisy User-generated Text*, pages 493–509. Association for Computational Linguistics.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021b. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Anders Johannsen, Héctor Martínez Alonso, and Barbara Plank. 2015. Universal Dependencies for Danish. In *International Workshop on Treebanks and Linguistic Theories (TLT14)*, page 157.

Nobuhiro Kaji and Masaru Kitsuregawa. 2014. Accurate word segmentation and POS tagging for Japanese microblogs: Corpus annotation and joint modeling with lexical normalization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 99–109, Doha, Qatar. Association for Computational Linguistics.

Andreas Kirkedal, Barbara Plank, Leon Derczynski, and Natalie Schluter. 2019. The lacunae of Danish natural language processing. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 356–362, Turku, Finland. Linköping University Electronic Press.

Matthias T Kromann, Line Mikkelsen, and Stine Kern Lynge. 2003. Danish Dependency Treebank. In *Proc. TLT*, pages 217–220. Citeseer.

Yijia Liu, Yi Zhu, Wanxiang Che, Bing Qin, Nathan Schneider, and Noah A. Smith. 2018. Parsing tweets into Universal Dependencies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 965–975, New Orleans, Louisiana. Association for Computational Linguistics.

Petter Mæhlum, Andre Kåsen, Samia Touileb, and Jeremy Barnes. 2022. Annotating Norwegian language varieties on Twitter for part-of-speech. In *Proceedings of the Ninth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 64–69, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Manuela Sanguinetti, Cristina Bosco, Lauren Cassidy, Özlem Çetinoğlu, Alessandra Teresa Cignarella, Teresa Lynn, Ines Rehbein, Josef Ruppenhofer, Djamé Seddah, and Amir Zeldes. 2020. Treebanking user-generated content: A proposal for a unified representation in Universal Dependencies. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5240–5250, Marseille, France. European Language Resources Association.

Kushagra Singh, Indira Sen, and Ponnurangam Kumaraguru. 2018. A Twitter corpus for Hindi-English code mixed POS tagging. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 12–17, Melbourne, Australia. Association for Computational Linguistics.

Xinyang Zhang, Yury Malkov, Omar Florez, Serim Park, Brian McWilliams, Jiawei Han, and Ahmed El-Kishky. 2022. Twhin-bert: A socially-enriched pre-trained language model for multilingual tweet representations. *arXiv preprint arXiv:2209.07562*.

Appendix

A Data Statement

The following dataset characteristics are outlined following Bender and Friedman (2018):

A. CURATION RATIONALE This dataset aims to provide high-quality, linguistically annotated data from contemporary Danish social media, in order to allow for analyses of how language use is evolving in these specialized domains, and how NLP methods can best be adapted to these changes.

B. LANGUAGE VARIETY The data consists of comments from TikTok videos collected in January 2023. The language covered is manually verified Danish (da-DK) with code-switching to English (en), and orthographic variations specific to the social media domain.

C. SPEAKER DEMOGRAPHIC Nothing specific is known about speaker demographics, as the data was scraped from 75 videos spanning different topics.

D. ANNOTATOR DEMOGRAPHIC Three Master's students, all native Danish speakers, one with previous experience in dataset creation for POS tagging. The annotators were paid for their efforts.

E. SPEECH SITUATION Comments under TikTok videos represent informal, written language produced largely spontaneously with the intent to address the video creator or express an opinion to other viewers.

F. TEXT CHARACTERISTICS The text contains domain-specific terms and abbreviations, some degree of typographical and orthographic errors as well as occasional ellipsis of sentence subject. Code-switching to English makes up 5% of tokens in the full dataset (development + test), though the dataset contains several additional tokens that exist with the same meaning in both English and Danish, e.g., *shit* and *like*.

G. RECORDING QUALITY N/A

H. OTHER N/A

I. PROVENANCE APPENDIX N/A

B Data Collection Details

B.1 Hashtags

Videos from the following 15 hashtags were scraped during data collection:

- #børn ("children")
- #danskememes ("Danish memes")
- #danskhumor ("Danish humor")
- #glædeligjul ("merry Christmas")
- #godtnytår ("happy new year")
- #gørdetselv ("do it yourself")
- #landsholdet ("the national team")
- #madlavning ("cooking")
- #mitarbejde ("my job")
- #morgenrutine ("morning routine")
- #parforhold ("relationships")
- #selvtak ("you're welcome")
- #sommerprojekt ("summer project")
- #tobiasrahim ("Tobias Rahim")
- #træning ("workout")

B.2 Deduplication Details

Figure 3 plots the number of tokens and their token-type ratios (TTR) after applying merge deduplication (Section 2.1) with threshold t.

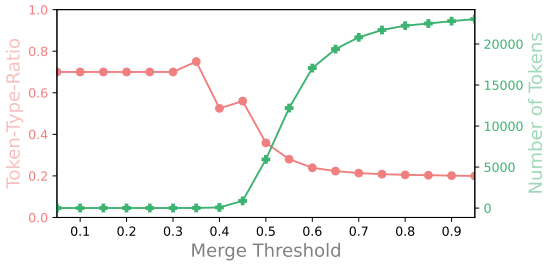


Figure 3: Deduplication using varying merge thresholds. A low t merges all comments into one, a high t contains more tokens and less token-type diversity.

C POS Tag Distribution

Figure 4 presents an overview of the POS tag distribution in DanTok compared to the English LINES, TWEEBANK2 and DDT.

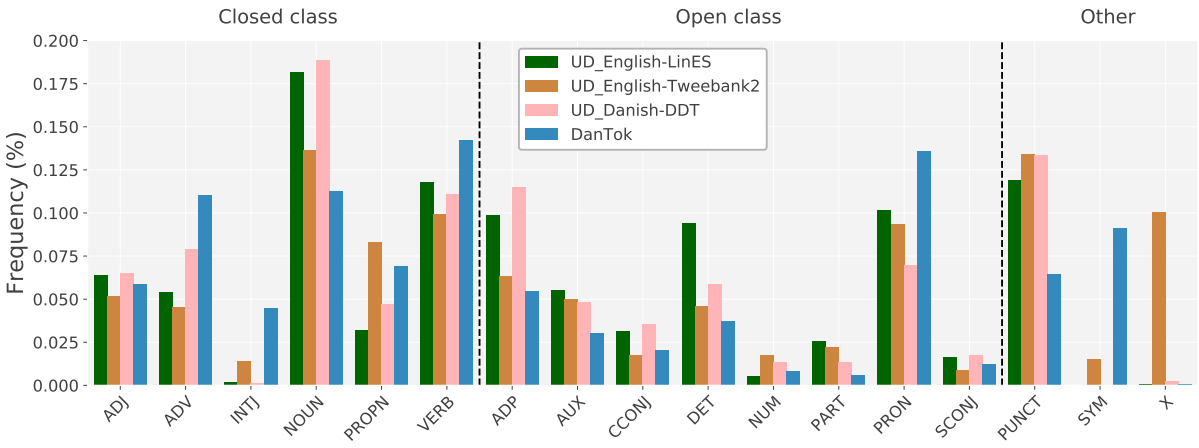


Figure 4: POS tag distribution in DanTok compared to the treebanks used for fine-tuning in Section 3.

Comparison of Current Approaches to Lemmatization: A Case Study in Estonian

Aleksei Dorkin
Institute of Computer Science
University of Tartu
aleksei.dorkin@ut.ee

Kairit Sirts
Institute of Computer Science
University of Tartu
kairit.sirts@ut.ee

Abstract

This study evaluates three different lemmatization approaches to Estonian—Generative character-level models, Pattern-based word-level classification models, and rule-based morphological analysis. According to our experiments, a significantly smaller Generative model consistently outperforms the Pattern-based classification model based on EstBERT. Additionally, we observe a relatively small overlap in errors made by all three models, indicating that an ensemble of different approaches could lead to improvements.

1 Introduction

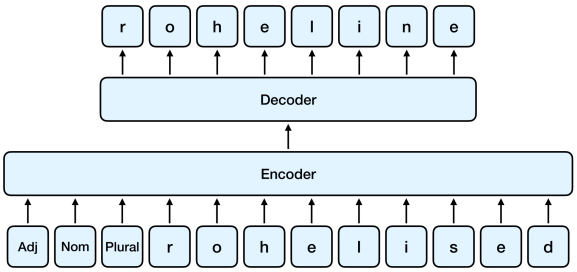
Recently, two different approaches have been adopted for model-based lemmatization. The Generative approach is based on encoder-decoder models and they generate the lemma character by character conditioned on the word form with its relevant context (Qi et al., 2020; Bergmanis and Goldwater, 2018). The Pattern-based approach treats lemmatization as a classification task (Straka, 2018), where each class is a transformation rule. When the correct rule is applied to a word-form, it unambiguously transforms the word-form to its lemma.

Our aim in this paper is to compare the performance of these two lemmatization approaches in Estonian. As a third approach, we also adopt the Estonian rule-based lemmatizer Vabamorf (Kaalep and Vaino, 2001). As all three approaches rely on different formalisms to lemmatization, we are also interested in the complementarity of these methods. The Generative approach is the most flexible, it has the largest search space and therefore it can occasionally result in hallucinating non-existing morphological transformations. On the other hand, the search space of the Pattern-based approach is much smaller as the model only has to correctly choose a single transformation class. However, if the required transformation is not present in the set of classes then the model is blocked from making the correct prediction. Similarly, the rule-based system can be highly precise but if it encounters a word that is absent from its dictionary the system can be clueless even if this word is morphologically highly regular.

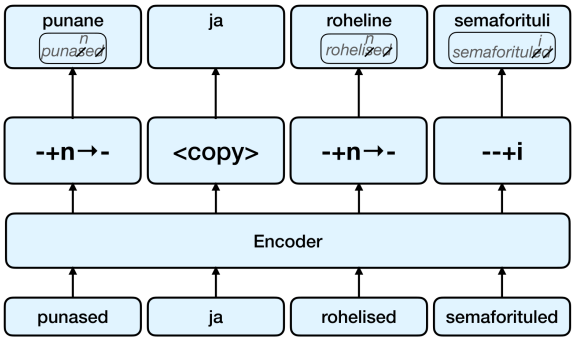
One problem with the recently proposed pattern-based approach implemented in the UDPipe2 is that the transformation rules mix the casing and morphological transformations. This means that for many morphological transformations there will be two rules in the ruleset—one for the lower-cased version of the word and another for the same word with the capital initial letter that needs to be lowered for the lemma—which increases the size of the ruleset considerably and thus artificially complicates the prediction task. Thus, in many cases a more optimal approach would be to treat casing separately from the lemmatization. Additionally, in the UD Estonian treebanks, lemmas include annotations of derivational and compounding processes marked by special symbols. However, these annotations are inconsistent in the data which confuses the models and also complicates the transformation rules. Thus, for our evaluation to be unaffected by these factors, we also train our models on the lowercased data with the special symbols removed.

Lemmatization models are commonly *token-based*, meaning that if the same word-form (with its relevant context) appears several times in the dataset, these repeating instances are kept in the data and thus the training and evaluation sets reflect the natural distribution of words. In contrast, for the morphological reinflection task, the custom has been to train *type-based* models, in which each lemma and morphological feature combination is presented to the model only once. We were interested in how well the type-based approach can work for lemmatization and thus we also experi-

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 280–285



(a) Generative approach.



(b) Pattern-based approach.

Figure 1: Schematic representations of the generative and pattern-based approaches.

mented with type-based models where appropriate.

In sum, our contributions in this paper are first comparing three different lemmatization approaches on two Estonian datasets with different domains with the goal of assessing the complementarity of these systems. Secondly, we investigate the effect of casing and special symbols as well as type- vs token-based training and evaluation for each comparison system.

2 Lemmatization approaches

This section gives a brief overview of the current approaches to lemmatization.

2.1 Generative approach

Generative lemmatization involves using a neural network to convert a word form, represented as a sequence of characters, into its lemma, also represented as a sequence of characters. The model is trained to predict the lemma in an auto-regressive manner, meaning that it makes predictions one character at a time based on the previously predicted characters. Commonly, generative lemmatization makes use of part of speech and morphological information as context (Qi et al., 2020). However, it is not necessarily limited to that. For example, Bergmanis and Goldwater (2018) propose using surrounding words, subword units, or characters as context for a given word form.

2.2 Pattern-based approach

In the Pattern-based lemmatization, the model assigns a specific transformation class to each word form, and then uses a predetermined rule to transform the word form to the lemma. The approach is not bound to any specific method of classification, or for that matter, representation of input features. For instance, in the UDPipe2 (Straka, 2018), the patterns are sequences of string edit operations, while the Spacy's lemmatizer uses an edit tree structure as a pattern (Müller et al., 2015).

2.3 Rule- and lexicon-based approaches

Rule-based approaches to lemmatization use various rule formalisms such as rule cascades or finite state transducers to transform the word form into lemma. For instance, the rule-based machine translation library Apertium also includes rule-based morphological analyzers for many languages (Khanna et al., 2021). For the Estonian language, there is a morphological analyzer called Vabamorf (Kaalep and Vaino, 2001). In the dictionary-based approach, the lemma of a word is determined by looking it up in a special dictionary. The dictionary may include word forms and their POS tags with morphological features, which can be used to identify the correct lemma. Such morphologial dictionaries include for instance Unimorph (McCarthy et al., 2020) and UD Lexicons (Sagot, 2018).

What these approaches have in common is that, intrinsically, they are not able to fully consider the context in which a given word form appears, which prevents them from disambiguating multiple candidates. So, for that purpose they have to rely on separate tools, such as Hidden Markov Models. They are also language-specific. The advantage, however, is that they are not dependent on the amount of training data, and can be quite precise.

3 Data

We use the Estonian Dependency Treebank (EDT) and the Estonian Web Treebank (EWT) from the Universal Dependencies collection version 2.10. The EDT comprises several genres such as newspaper texts, fiction, scientific articles, while the EWT is composed of texts from internet blogs and forums. The statistics of both datasets are given in Table 1.

	train	dev	test
EDT # of sentences	24633	3125	3214
EDT # of tokens	344953	44686	48532
EWT # of sentences	4579	833	913
EWT # of tokens	55143	10012	13176

Table 1: Number of sentences and tokens per split in Estonian Dependency Treebank and Estonian Web Treebank as of version 2.10.

4 Implementation

For the Generative approach, we adopted the neural transducer by Wu et al. (2020), previously used for morphological reinflection. Neural transducer is a character-level transformer, which takes individual characters of a word form and morphological tags as input, and outputs the resulting lemma character-by-character.

For the Pattern-based model we adopted an approach similar to UDpipe2 (Straka, 2018). We used a transformer-based token classification model by fine-tuning EstBERT (Tanvir et al., 2020) to predict the correct transformation class (form → lemma) for every token in a sentence. Our model uses HuggingFace (Wolf et al., 2020) TokenClassification implementation. Moreover, we reuse the code to generate transformation classes from UDpipe2.[1]

For the rule-based approach, we adopted the Estonian rule-based morphological analyzer Vabamorf (Kaalep and Vaino, 2001). We used Vabamorf via EstNLTK, which is a library that provides an API to various Estonian language technology tools (Orasmaa et al., 2016). We utilized Vabamorf's HMM-based disambiguation capabilities to output a single lemma for each token.

5 Results

Tables 2 and 3 show the results on the EDT and EWT validation sets respectively. Overall, the Generative model (in the token-based training setting, see below) outperforms both the Pattern-based model and the rule-based Vabamorf on both datasets.

The first column (Original) in Table 2 shows results on the EDT data in its original case sensitive form and including special symbols marking derivation and compounding. The second column (No Sym) shows the results of models trained on lowercased data with special symbols removed. All approaches show a noticeable improvement in accuracy in the simplified environment. Although the improvement with the Pattern-based model is the smallest, it has the largest implications—ignoring casing and removing special symbols halves the number of transformation classes.

	Original	No Sym	Type Eval
Gen Token	95.49	97.59	97.61
Gen Type	91.55	95.64	95.10
Pattern-based	95.04	96.34	–
Vabamorf	87.78	91.66	–
Vabamorf Oracle	99.31	99.47	–

Table 2: Lemmatization accuracy on the EDT validation set. Original: unaltered EDT, No Sym: lowercased EDT with special symbols removed, Type Eval: evaluation on distinct word types with No Sym setting.

Trained on	EWT	EDT
Gen Token	95.88	96.28
Gen Type	94.63	95.97
Pattern-based	95.02	87.97
Vabamorf	91.75	91.74
Vabamorf Oracle	96.98	96.98

Table 3: Lemmatization accuracy on the EWT validation set. The first column contains results for models trained on EWT, the results for models trained on EDT are shown in the second column.

The top part of the Tables 2 and 3 compare the results of the Generative model trained on word tokens and word types. Additionally, the last column of the Table 2 also shows the evaluation on unique types of both the token-based and type-based models trained in the No Sym setting. The Generative model trained on word tokens always performs better than the model trained on unique word types even when evaluated on word types. We conclude that there does not seem to be any disadvantages to token-based training.

In Table 3, EWT validation set is evaluated in two settings. The first column shows the results of the in-domain models trained on the EWT train set, the results in the second column are obtained

[1] https://github.com/ufal/udpipe/blob/udpipe-2/udpipe2_dataset.py

with the out-of-domain models trained on the EDT train set. We observe that the Generative models perform well in the cross-domain environment, and outperform the model trained on the EWT train set. Meanwhile, the Pattern-based model trained on the EDT shows a significant drop in performance when evaluated on EWT. Vabamorf also demonstrates a degraded performance on EWT.

The last row in both Tables 2 and 3 show the performance of the Vabamorf in the oracle mode, in which case the prediction is considered correct if the true lemma appears in the list of generated candidates. We observe a significant improvement in the accuracy of Vabamorf in the oracle mode. This means that a large chunk of errors made by the rule-based approach is the result of poor disambiguation, rather than incorrect morphological analysis.

In addition to comparing the performance of different approaches, we are interested in whether there is any complementarity in the errors made by models based on different approaches. Figure 2 presents a Venn diagram of the token-level errors made by each system. We note that the area of the intersection of all three models is relatively small, meaning that the number of words where all models make an error is quite small, suggesting that different approaches can complement each other in an ensemble setting.

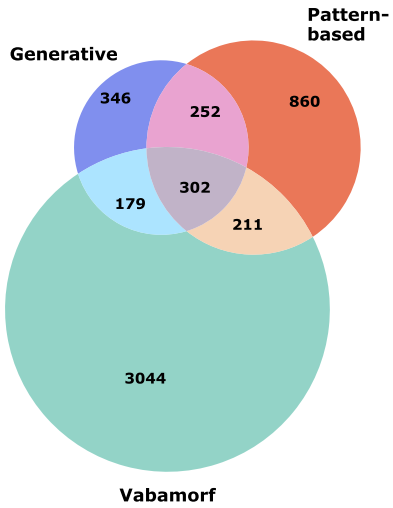


Figure 2: Venn diagram of the token-level lemmatization errors made by each model on the EDT validation set.

6 Discussion

Because the UDPipe2's pattern-based approach was highly successful in the Sigmorphon 2019 shared task (Straka et al., 2019), we expected it to perform well also in our case, especially because instead of the frozen BERT weights used in the UDPipe2, we fine-tuned the full model. However, the best shared task lemmatization scores for the Estonian language were obtained with the generative contextual lemmatizer by Bergmanis and Goldwater (2018), which perhaps explains the success of the Generative model also in our experiments. When analyzing the errors made by each three approaches, we can see that the set of errors where all models overlap is relatively small (302 out of 5194, 5.8%), which suggests that different approaches can potentially compensate for each other and thus an ensemble of different methods can be useful.

The rule-based Vabamorf made the largest number of errors. However, when we evaluated it in the oracle mode on EDT, it covered the vast majority of correct answers. This implies that Vabamorf could gain a lot from a better disambiguator than the current HMM-based one. This was not the case for EWT which, being a web treebank, contains more word forms (such as neologisms, more recent loanwords, and so on) missing from the Vabamorf's lexicon. Thus, while Vabamorf can be a good solution for formal and grammatically correct Estonian, it is less suitable for more noisy web texts.

The approach to creating transformation rules suggested by the developers of UDpipe may output equivalent rules, i.e., when applying these rules to a surface form, the result is identical. We noticed that the Pattern-based model is able to identify such cases. This means that an incorrectly predicted *label* can result in a correct lemma. For example, two rules `↓0;d¦---+m+a` and `0;d¦-+m→-` transform the third person plural present tense form into the corresponding -ma infinitive (*vabandavad* → *vabandama* "to apologize"). The difference between these rules is that the former rule removes three last letters and adds *ma*-suffix, while the latter removes the last letter, and then replaces the existing letter preceding the existing *a* with *m*. We suggest that such a peculiarity may be used to probe language models for morphological knowledge.

The five most common rules are shown in Table 4. The most common rule is the "do-nothing" rule, which accounts for more than half of the occurrences in the EDT train set. The next three most common rules with smaller but still considerable frequency involve removing suffixes of varying

%	Rule	Description
54.1	↓0;d¦	Do nothing
8.3	↓0;d¦-	Remove the last letter
5.2	↓0;d¦--	Remove two last letters
3.4	↓0;d¦---	Remove three last letters
3.3	↓0;d¦-+m+a	Replace the last letter with *ma*

Table 4: Top 5 most common transformation rules present in the train split of the EDT dataset.

length. The last rule fitting into our top-5 list is specific to verbs, replacing the last character with the lemma suffix for verbs. The total set contains a very long tail of transformation rules that appear only a few times or just once, such as rules corresponding to the transformation of infrequently used suppletive forms.

7 Conclusion

We compared three lemmatization approaches on two Estonian datasets from different domains and found that on both datasets the Generative encoder-decoder approach trained from scratch outperforms both the rule-based Vabamorf as well as the Pattern-based approach fine-tuned from a large pre-trained language model. We observed complementary error patterns for each three approaches, which suggests that ensembling techniques can take advantage of the complementary strengths of each individual approach.

Acknowledgments

This research was supported by the Estonian Research Council Grant PSG721.

References

Toms Bergmanis and Sharon Goldwater. 2018. Context sensitive neural lemmatization with Lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1391–1400, New Orleans, Louisiana. Association for Computational Linguistics.

Heiki-Jaan Kaalep and Tarmo Vaino. 2001. Complete morphological analysis in the linguist's toolbox. *Congressus Nonus Internationalis Fenno-Ugristarum Pars V*, pages 9–16.

Tanmai Khanna, Jonathan N Washington, Francis M Tyers, Sevilay Bayatlı, Daniel G Swanson, Tommi A Pirinen, Irene Tang, and Hèctor Alòs i Font. 2021. Recent advances in apertium, a free/open-source rule-based machine translation platform for low-resource languages. *Machine Translation*, 35(4):475–502.

Arya D. McCarthy, Christo Kirov, Matteo Grella, Amrit Nidhi, Patrick Xia, Kyle Gorman, Ekaterina Vylomova, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, Timofey Arkhangelskiy, Nataly Krizhanovsky, Andrew Krizhanovsky, Elena Klyachko, Alexey Sorokin, John Mansfield, Valts Ernštreits, Yuval Pinter, Cassandra L. Jacobs, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2020. UniMorph 3.0: Universal Morphology. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3922–3931, Marseille, France. European Language Resources Association.

Thomas Müller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. 2015. Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274, Lisbon, Portugal. Association for Computational Linguistics.

Siim Orasmaa, Timo Petmanson, Alexander Tkachenko, Sven Laur, and Heiki-Jaan Kaalep. 2016. Estnltk - nlp toolkit for estonian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Benoît Sagot. 2018. A multilingual collection of CoNLL-U-compatible morphological lexicons. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Milan Straka, Jana Straková, and Jan Hajic. 2019. Udpipe at sigmorphon 2019: Contextualized embeddings, regularization with morphological categories, corpora merging. *SIGMORPHON 2019*, page 95.

Hasan Tanvir, Claudia Kittask, and Kairit Sirts. 2020. Estbert: A pretrained language-specific BERT for estonian. *CoRR*, abs/2011.04784.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara

Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2020. Applying the transformer to character-level transduction. *CoRR*, abs/2005.10213.

Generating Errors: OCR Post-Processing for Icelandic

Atli Jasonarson, Steinþór Steingrímsson, Einar Freyr Sigurðsson,
Árni Davíð Magnússon, Finnur Ágúst Ingimundarson
The Árni Magnússon Institute for Icelandic Studies, Iceland
{atli.jasonarson, steinthor.steingrimsson,
einar.freyr.sigurdsson, arni.d.magnusson}@arnastofnun
fai@hi.is

Abstract

We describe work on enhancing the performance of transformer-based encoder-decoder models for OCR post-correction on modern and historical Icelandic texts, where OCRed data are scarce. We trained six models, four from scratch and two fine-tuned versions of Google's ByT5, on a combination of real data and texts populated with artificially generated errors. Our results show that the models trained from scratch, as opposed to the fine-tuned versions, benefited the most from the addition of artificially generated errors.

1 Introduction

Optical Character Recognition (OCR) is used to digitize texts by converting scanned documents into machine-readable text. Unfortunately, OCR errors are prevalent, particularly when it comes to old texts, where data tends to be scarce, and post-correction is often required to improve the extracted texts' accuracy (e.g. Nguyen et al. 2021).

Transformer-based encoder-decoder models have been shown to be effective in various natural language processing tasks, including machine translation (Vaswani et al., 2017; Chen et al., 2018) and text summarization (Garg et al., 2021). In this study, we investigate the use of such models for OCR post-correction under scarce data condition, as a sequence-to-sequence problem, similar to how neural machine translation (NMT) systems approach the problem of translation. To address the lack of resources available for training models when dealing with OCRed texts, we propose the use of artificially generated errors to improve the performance of the models, which has been shown to be an effective way of generating data for text correction (Kasewa et al., 2018). The main contribution of this study is an examination of the effectiveness of using artificially generated errors to improve the performance of transformer-based encoder-decoder models for OCR post-correction when data scarcity is a limiting factor. Furthermore, we publish our best performing models under the Apache 2.0 license.[1]

The paper is structured as follows. Section 2 discusses related work while Section 3 describes the dataset and error generation methods used in this study. Section 4 presents the proposed models and training methods. Section 5 presents the experimental results and analysis, Section 6 discusses limitations and future work, and finally Section 7 concludes.

2 Related Work

Previously, Daðason et al. (2014) developed a tool for post-processing Icelandic 19th century texts based on an error model containing statistical information on word and character errors and an n-gram language model. Their tool correctly identifies and corrects 52.9% of errors in their evaluation set.

Poncelas et al. (2020) report a 63% error correction rate with their OCR post-processing tool on an English text from the 18th century. They used a scoring system based on string-similarity to find possible substitutions for perceived errors and a language model to evaluate the edited sentences.

Richter et al. (2018) use a hidden Markov model alongside a modified version of the Viterbi algorithm and a dictionary to decode OCRed texts in Faroese into a hypothetical corrected version. They reduced the word error rate of 7.8% from the OCR base process to 5.4%.

[1] Models available at the following URLs: http://hdl.handle.net/20.500.12537/271, http://hdl.handle.net/20.500.12537/309, https://huggingface.co/atlijas/byt5-is-ocr-post-processing-modern-texts, https://huggingface.co/atlijas/byt5-is-ocr-post-processing-old-texts.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 286–291

Original	Corrected	Frequency
p	*þ*	2,779
i	*í*	1,141
li	*h*	247
rn	*m*	166
m	*rn*	77

Table 1: Examples of the extracted errors.

3 Data

We used a combination of real OCRed texts, processed by ABBYY FineReader,[2] and digital texts not scanned with OCR, the latter of which were populated with artificially generated errors, for training and evaluating our OCR post-correction models. The evaluation data solely comprised real OCRed texts, alongside their manually corrected counterparts, which were used to ensure that the models' performance is reflective of real-world OCR output. The ground truth (GT),[3] i.e. the data from which the errors were extracted, consists of around 375k tokens from 80 texts published between 1874 and 1913, which were manually corrected, while the data used in training and validation, which include the OCRed texts as well as the texts populated with the artificial errors, amount to roughly 9.2M tokens.

The data into which the artificial errors were inserted were taken from the Icelandic Gigaword Corpus (IGC; Steingrímsson et al. 2018) and the Icelandic Text Archive (ITA).[4] Their publication dates range from the late 18th century to the early 21st century, with roughly 40% of them having been published between 1830 and 1920.

Overall, the training data consist of 7.8M tokens, whereas the validation set, which is approximately 15% of the whole dataset, consists of around 1.4M tokens. The evaluation set, totalling 44k tokens, is composed of manually corrected texts. All datasets contain texts from different eras, including texts from the 19th century and the early 20th century, as well as texts from the last two decades of the 20th century. It should be noted that none of the data are based on texts printed in Gothic font, which has been reported to be harder to recognize than other fonts (Furrer and Volk, 2011; Drobac et al., 2017). The evaluation set is divided into two parts, with 26k tokens being from modern texts and 18k tokens from texts from the 19th century and the early 20th century. This allows for an evaluation of the model's performance on different types of texts and OCR errors, which is crucial to ensure that the model is robust and generalizable.

It is important to note that the dataset used in this study is relatively small in size. One of the reasons is the scarcity of available corrected OCRed texts. Additionally, we observed that too large a proportion of modern texts in the training set resulted in the models over-generalizing and changing historical spellings to modern spellings. However, we aim for diplomatic transcription, preserving the original spelling. Therefore, we ensured that the dataset included texts from different eras while also avoiding over-generalization and alteration of historical spellings by limiting the amount of modern texts into which we inserted artificial errors.

	Model 1	Model 2
embeddings size	512	512
ffn embeddings	2,048	2,048
attention heads	4	4
encoder layers	5	5
decoder layer	5	5
tokenizer	WordPiece	SentencePiece
vocab. size	3,000	3,000

Table 2: The architecture of the two models trained from scratch.

3.1 Extracting the Errors

The extraction of errors from the manually corrected OCRed texts was performed by analysing the 375k token dataset. The data were manually aligned, and then a line-by-line comparison was conducted between the OCRed texts and their corresponding manually corrected texts using Python's SequenceMatcher. In the process of extracting errors, tokens were considered to be the same if they shared the same index in a given line and had a similarity score greater than 0.66.[5] This twofold restriction, taking into account both index

[2] https://pdf.abbyy.com/

[3] The GT is a product of the project *Language Change and Linguistic Variation in 19th-Century Icelandic and the Emergence of a National Standard*, led by Ásta Svavarsdóttir at the Árni Magnússon Institute for Icelandic Studies (e.g. Svavarsdóttir et al. 2014).

[4] https://clarin.is/en/resources/textarchive/

[5] Calculated by finding "[...] the longest continuous matching subsequence that contains no "junk" elements", see: https://docs.python.org/3/library/difflib.html.

	Older texts				Modern texts			
	OCR	Model 1	Model 2	ByT5 (5 ep.)	OCR	Model 1	Model 2	ByT5 (1 ep.)
chrF	94.79	94.80	96.00	96.22	95.27	95.52	95.75	96.09
BLEU	97.19	97.19	98.24	98.54	97.73	97.63	98.06	98.24
WER	6.49%	7.56%	4.22%	3.25%	5.52%	5.73%	4.58%	4.56%
WERR	Ø	-16.37%	35.04%	49.96%	Ø	-3.80%	17.02%	17.37%
CER	1.39%	1.79%	1.14%	0.92%	1.17%	1.63%	1.43%	1.41%
CERR	Ø	-28.53%	18.34%	33.83%	Ø	-38.58%	-21.34%	-20.34%

Table 3: Our models trained on the GT compared to the base output from the OCR process. WER(R) = Word Error Rate (Reduction), CER(R) = Character Error Rate (Reduction).

and similarity, acted as a confidence threshold to ensure that the identified tokens were different versions of the same intended token.

The differences between the tokens, specifically focusing on character or character n-gram substitutions, such as *rn*→*m* and *þ*→*p*, were extracted as OCR errors. In total, 2,644 such error types were extracted, which were then filtered down to the 600 errors that occurred more than three times in the dataset. In addition, the frequency of each error was recorded, which allowed for the implementation of a weighting system during the artificial error generation process, ensuring that the errors were distributed in a way that somewhat reflected their real-world frequency. Examples of extracted errors are shown in Table 1.

Error pairs that consist of an original and corrected string length 1 (character count) comprise around 40% of the error set. An example of this is the erroneous ***p**essi* for ***þ**essi* 'this'. In 30% of error pairs the original has a length of 2 and the corrected a length of 1, such as ***rn**aður* for ***m**aður* 'man', and in about 15% of them the length of both is 2, e.g. *g**db**ur* for *g**óð**ur* 'good'. The total number of errors in the dataset amounts to 27,369.

3.2 Inserting the errors

To create the training dataset, we gathered texts from IGC and the ITA. Texts ranging from the late 18th century to the 21st century were collected to provide a diverse set of texts for model training.

Error types, as extracted and described in Section 3.1, were then inserted into the training dataset by randomly replacing characters or character n-grams via a lookup table. Whitespace was also removed from between tokens and added into the tokens at random. The artificial errors were inserted randomly, with the frequency of error types based on the frequency in the GT in order to mimic the distribution of errors that occur in OCR output. This way, more frequent errors in the GT were made to appear more frequently than other errors in the training dataset. However, to prevent the same errors from appearing excessively often in the dataset, we used the $\log_{10}\text{frequency}$ of the errors.

4 Models

In total, six models were trained. Two of them follow the architecture of model 1, laid out in table 2, two of them follow the architecture of model 2 in the same table, and the others are a fine-tuned version of ByT5-base[6] (Xue et al., 2022), a token-free transformer model that operates directly on UTF-8 encoded bytes and is trained on mC4, a multilingual corpus, which consists of texts in 101 languages, including Icelandic (Xue et al., 2021). The models are all encoder-decoder transformer models.

For every pair of the models, one was trained on the 375k tokens in the GT, and the other one on the whole dataset, around 7.8M tokens. This was done to study the artificially generated errors' impact on the models' output. We experimented with various hyperparameter configurations, evaluating the models we trained from scratch on the validation sets, and these specific configurations resulted in the highest performance.

It is well established that transformer models require large amounts of data to be trained effectively. In this study, our GT had a limited number of examples, which likely contributed to the

[6]Note that the ByT5 model was trained for five epochs, resulting in five different models. The one trained for one epoch performed the best on modern texts while the one trained for five epochs performed the best on older texts. We report on these two ByT5 models in Table 3.

poorer performance of the models trained from scratch on the smaller dataset, in some instances even performing worse than the base OCR process.

4.1 Tokenizers

As seen in Table 2, the two models trained from scratch use different tokenizers, both of which are based on subword tokenization algorithms. Model 1 uses WordPiece (Song et al., 2021) and model 2 uses SentencePiece (Kudo and Richardson, 2018). As mentioned before, the ByT5 model operates directly on UTF-8 encoded bytes.

Different tokenization algorithms can have an impact on a given task. SentencePiece and WordPiece can produce different subword units for the same text, which might affect the models' ability to capture language-specific nuances and patterns. It is possible that the choice of tokenizer had some impact on their performance. However, further research would be needed to determine the specific effects of the tokenizer choice on OCR error correction.

5 Results

The six models we trained for post-processing of OCRed texts were applied to modern and historical texts to measure the impact and viability of using artificial errors to improve such models when available data are scarce. The results of these models were compared to the base output from the OCR process using four metrics: chrF (Popović, 2015) and BLEU (Papineni et al., 2002), character error rate (CER) and word error rate (WER). BLEU score is calculated by comparing texts on a word-level, while chrF score is calculated on a character-level and can be more accurate for inflected languages (Dowling et al., 2020).

Table 3 shows the results of our models trained on the GT compared to the base OCR output. Model 2 and the ByT5 model show moderate improvements for older texts, while model 1 performs similarly or worse than the base OCR output. Generally, the models do worse on the modern texts, as opposed to the historical ones, when only trained on the GT, which is to be expected as the GT solely consists of historical texts.

Table 4 shows the results of our models trained on the whole dataset compared to the base OCR output. The models all show substantial improvements compared to the models only trained on the GT, which suggests that the artificial errors have something to offer. Furthermore, the difference between word error rate reduction (WERR) for the different text types was less than for the models only trained on the GT.

Note that while the models generally perform better on the historical texts, the addition of artificial errors improve their performance proportionally more on the modern ones. This could stem from the fact that the artificially-erroneous dataset includes modern texts, while the GT does not.

When evaluating the models on modern texts, we found that they were less capable in reducing errors in modern texts than in historical texts. This could be due to the fact that the GT only comprised historical data, suggesting that using solely historical OCRed texts is not a viable approach when designing an OCR post-processing tool for modern texts. The lower error rate reduction (ERR) on the modern texts presumably also stems from the higher base score on the modern texts, as opposed to the base score of the historical ones, leaving less room for improvement.

6 Limitations and Future Work

The cost of manually correcting OCR output is high, making it difficult to obtain a larger dataset for training. This has a direct impact on the ability of the models to perform well on a wider range of texts and OCR errors.

The models have the unwanted tendency to adapt to modern spellings when using a large amount of modern texts populated with artificial errors. This could lead to the alteration of historical spellings, which is not in line with our objectives, to produce diplomatic transcriptions. To mitigate this risk, more corrected texts are needed for the period of texts being OCRed.

Moreover, the use of artificially generated errors to enhance the performance of the models may not fully capture the complexity and diversity of real-world OCR errors. Future studies may benefit from incorporating a more diverse range of error types and more realistic error generation methods.

We are interested in investigating optimal methods for generating realistic errors to use in training the models. As previously mentioned, the artificial errors used in this study were generated by randomly inserting errors that were extracted from the GT into other texts. However, there may be more effective methods for generating errors that better

	Older texts				Modern texts			
	OCR	Model 1	Model 2	ByT5 (5 ep.)	OCR	Model 1	Model 2	ByT5 (1 ep.)
chrF	94.79	96.84	96.84	96.73	95.27	96.83	96.86	96.7
BLEU	97.19	98.45	98.79	98.65	97.73	98.45	98.64	98.57
WER	6.49%	4.95%	3.08%	2.92%	5.52%	4.52%	3.60%	3.15%
WERR	Ø	23.79%	52.60%	55.07%	Ø	18.00%	34.67%	42.97%
CER	1.39%	1.03%	0.73%	0.90%	1.17%	1.06%	1.0%	1.15%
CERR	Ø	26.29%	47.55%	35.12%	Ø	10.01%	15.20%	1.93%

Table 4: Our models trained on the whole dataset compared to the base output from the OCR process.

simulate real-world OCR errors. By finding and implementing these methods, the performance of OCR error correction models could be further improved. Furthermore, it could be beneficial to explore different architectures or different data augmentation techniques, such as including multiple versions of the same texts. It should also be noted that our evaluation dataset was rather small, and further testing on larger datasets may provide a more robust evaluation of the models.

7 Conclusion

Our findings demonstrate that while fine-tuning pre-trained models on smaller datasets is an effective approach to improving the performance of OCR error correction models, it is possible to achieve comparable results by training an encoder-decoder transformer model from scratch. Model 2, which was trained from scratch, emerged as the best performer in our study, achieving a 52.60% word error rate reduction (WERR) and a 47.55% character error rate reduction (CERR) on the historical texts, and a word error rate reduction of 34.67% and a character error rate reduction of 15.20% on the modern texts, see Table 4.

These results indicate that with proper architectural design, it is possible to train effective OCR error correction models without relying on pre-trained models or large datasets.

However, the use of artificially generated errors in the training process was found to be effective in countering the challenges posed by data scarcity.

The fact that the models' performance improved proportionally more on the modern texts after the introduction of the artificial errors, which were by and large inserted into modern texts, indicates that in order to train a designated OCR post-processing tool for modern texts, a dataset consisting of modern texts is needed.

Acknowledgements

This work is supported by the Language Technology Programme for Icelandic 2019–2023, funded by the Icelandic government, and the Icelandic Infrastructure Fund (grant no. 200336-6101).

References

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86, Melbourne, Australia.

Jón Friðrik Daðason, Kristín Bjarnadóttir, and Kristján Rúnarsson. 2014. The Journal *Fjölnir* for Everyone: The Post-Processing of Historical OCR Texts. In *Proceedings of Language Resources and Technologies for Processing and Linking Historical Documents and Archives – Deploying Linked Open Data in Cultural Heritage*, pages 65–62, Reykjavik, Iceland.

Meghan Dowling, Sheila Castilho, Joss Moorkens, Teresa Lynn, and Andy Way. 2020. A human evaluation of English-Irish statistical and neural machine translation. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 431–440, Lisbon, Portugal.

Senka Drobac, Pekka Sakari Kauppinen, and Bo Krister Johan Linden. 2017. OCR and post-correction of historical Finnish texts. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, page 70–76, Gothenburg, Sweden.

Lenz Furrer and Martin Volk. 2011. Reducing OCR Errors in Gothic-Script Documents. In *Proceedings of the Workshop on Language Technologies for Digital Humanities and Cultural Heritage*, pages 97–103, Hissar, Bulgaria.

Apar Garg, Saiteja Adusumilli, Shanmukha Yenneti, Tapas Badal, Deepak Garg, Vivek Pandey, Abhishek Nigam, Yashu Kant Gupta, Gyan Mittal, and Rahul Agarwal. 2021. NEWS Article Summarization with Pretrained Transformer. In *Advanced Computing: 10th International Conference, IACC 2020, Panaji, Goa, India, December 5–6, 2020, Revised Selected Papers, Part I 10*, pages 203–211. Springer.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. Wronging a Right: Generating Better Errors to Improve Grammatical Error Detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4977–4983, Brussels, Belgium.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium.

Thi Tuyet Hai Nguyen, Adam Jatowt, Mickael Coustaty, and Antoine Doucet. 2021. Survey of Post-OCR Processing Approaches. *ACM Computing Surveys*, 54(6).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania.

Alberto Poncelas, Mohammad Aboomar, Jan Buts, James Hadley, and Andy Way. 2020. A Tool for Facilitating OCR Postediting in Historical Documents. In *Proceedings of LT4HALA 2020 – 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 47–51, Marseille, France.

Maja Popović. 2015. CHRF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal.

Caitlin Richter, Matthew Wickes, Deniz Beser, and Mitch Marcus. 2018. Low-resource Post Processing of Noisy OCR Output for Historical Corpus Digitisation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 2331–2339, Miyazaki, Japan.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. 2021. Fast WordPiece Tokenization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2089–2103, Online and Punta Cana, Dominican Republic.

Steinþór Steingrímsson, Sigrún Helgadóttir, Eiríkur Rögnvaldsson, Starkaður Barkarson, and Jón Guðnason. 2018. Risamálheild: A Very Large Icelandic Text Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, LREC 2018, pages 4361–4366, Miyazaki, Japan.

Ásta Svavarsdóttir, Sigrún Helgadóttir, and Guðrún Kvaran. 2014. Language resources for early Modern Icelandic. In *Proceedings of Language Resources and Technologies for Processing and Linking Historical Documents and Archives – Deploying Linked Open Data in Cultural Heritage*, pages 19–25, Reykjavik, Iceland.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5999–6009, Long Beach, California.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online.

Generation of Replacement Options in Text Sanitization

Annika Willoch Olstad
Language Technology Group
University of Oslo
Oslo, Norway
annikaol@ifi.uio.no

Anthi Papadopoulou
Language Technology Group
University of Oslo
Oslo, Norway
anthip@ifi.uio.no

Pierre Lison
Norwegian Computing Center
Oslo, Norway
plison@nr.no

Abstract

The purpose of text sanitization is to edit text documents to mask text spans that may directly or indirectly reveal personal information. An important problem in text sanitization is to find less specific, yet still informative replacements for each text span to mask. We present an approach to generate possible replacements using a combination of heuristic rules and an ontology derived from Wikidata. Those replacement options are hierarchically structured and cover various types of personal identifiers. Using this approach, we extend a recently released text sanitization dataset with manually selected replacements. The outcome of this data collection shows that the approach is able to suggest appropriate replacement options for most text spans.

1 Introduction

Most texts contain Personally Identifiable Information (PII), which is information that can be used to directly or indirectly identify an individual. This raises privacy problems, as privacy frameworks such as GDPR (GDPR, 2016) enshrine the right of each individual to control the availability and sharing of their personal information.

Although full, GDPR-compliant anonymization is difficult to achieve (Weitzenboeck et al., 2022), it is often desirable to apply *text sanitization* techniques to mask (i.e. remove or replace) PII from a given text and thereby conceal the identity of the persons referred to in the document. Those PII can either correspond to *direct identifiers* (e.g. names, addresses, telephone numbers or social security identifiers) or take the form of so-called *quasi-identifiers* which are information that do not identify a person when seen in isolation, but may do so when combined together (Elliot et al., 2016). Examples of quasi-identifiers are postal codes, gender, age, employer or profession.

Most text sanitization approaches operate by (1) detecting text spans that convey PII and (2) replacing them with a default string such as '***' or a black box (Lison et al., 2021; Pilán et al., 2022). However, this considerably reduces the utility of the sanitized document. An alternative is to replace the detected text spans with more general values that are less risky from a privacy perspective, but remain more informative than a default string. For instance, *Drammen* may be replaced by *[city in Norway]*, *Telenor* by *[telecommunications company]* and *February 5, 2023* by *[2023]*.

The paper makes two contributions:

- An approach (illustrated in Figure 1) that generates suitable generalization options for different types of PII, based on heuristic rules and an ontology derived from Wikidata.
- *WikiReplace*, an extension of the dataset from Papadopoulou et al. (2022a) in which human annotators select for each text span the most suitable replacement among the possible alternatives produced by the above approach. The dataset is made freely available[1].

The paper focuses on the specific problem of generating replacement choices for text spans expressing PII. The problem of how those text spans should be detected and classified lies therefore outside the scope of this paper. This span detection can be implemented using various types of sequence labelling models, as detailed in Dernoncourt et al. (2017); Lison et al. (2021); Pilán et al. (2022)

The rest of the paper is constructed as follows. Section 2 describes previous work in this task, while Section 3 presents the replacement approach. In Section 4 we present the dataset and evaluate its quality. We conclude in Section 5.

[1] https://github.com/anthipapa/bootstrapping-anonymization

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 292–300

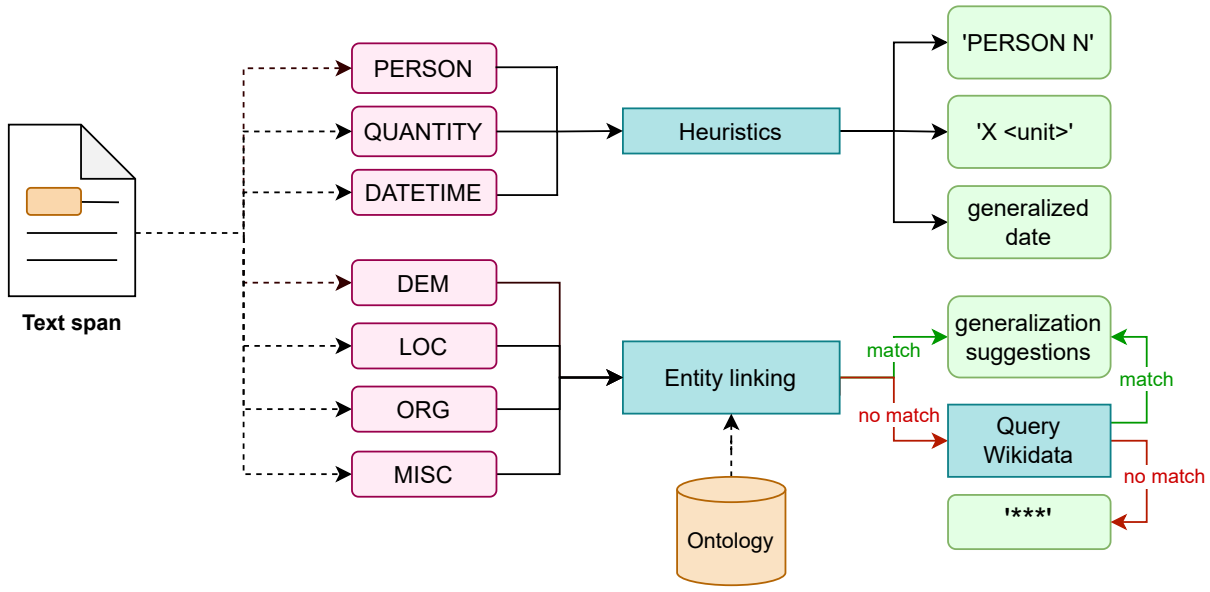


Figure 1: Generation of replacement options for text spans. Depending on the entity type, the replacements are produced using either heuristics or the Wikidata-derived ontology.

2 Related work

How to replace text spans expressing PII has been investigated in both Natural Language Processing (NLP) and in Privacy-Preserving Data Publishing (PPDP). Most NLP approaches (Bråthen et al., 2021; Dernoncourt et al., 2017; Pilán et al., 2022; Papadopoulou et al., 2022b) simply replace the detected text spans by a default string or a black box. Some alternatives include replacing text spans by pseudonyms (Dalianis, 2019; Volodina et al., 2020) or synthetic surrogates (Carrell et al., 2012). In the medical domain, identified names in patient records can be replaced with random names from a list (Dalianis, 2019). One can also rely on lexical substitution, in which target words are replaced with similar lexical entities, e.g. a synonym or hypernym (McCarthy and Navigli, 2007). This substitution can be implemented using various neural language models (Zhou et al., 2019; Arefyev et al., 2020).

Within the field of PPDP, the C-sanitize approach (Sánchez and Batet, 2016) frames the replacement of quasi-identifiers through an automatic sanitization process that mimics manual sanitization. It replaces identifiers with suitable generalizations, selected from a knowledge base, and an a parameter that can be adjusted to trade between privacy protection and data utility. t-plausibility (Anandan et al., 2012) generalizes identifiers so that at least t documents are derived through specialization of the generalized terms.

3 Generation of potential replacements

We follow the categorization of text spans expressing PII detailed in Pilán et al. (2022):

PERSON Names of people.

CODE Numbers and identification codes.

LOC Places and locations.

ORG Names of organizations.

DEM Demographic attributes of a person, such as job title, education, ethnicity or language.

DATETIME Specific date, time or duration.

QUANTITY Quantity, including percentages or monetary values.

MISC Every other type of information not belonging to the categories above.

Entities of type PERSON, QUANTITY and DATETIME are replaced using the heuristics in Section 3.1. Entities of type LOC, ORG, DEM and MISC are replaced by generalizations found in the ontology through entity linking, as described in Section 3.2.1. If no generalization can be found in the ontology, the system queries Wikidata directly. If this query does not return any generalization, '***' is returned. As entities of type CODE cannot be generalized, they are replaced by '***'.

3.1 Rule-based generation

We developed a set of heuristic rules to generalize entities of type PERSON, QUANTITY and DATETIME:

- PERSON entities are replaced by the text span [*PERSON N*], where N is an integer:

  "Ada Lovelace" → [*PERSON 1*]

  Terms that are found to refer to the same individual (based on e.g. coreference resolution) are assigned to the same integer.

- QUANTITY entities are replaced by X followed by the unit of measurement if applicable:

  "23 €" → [*X €*].

- DATETIME entities are generalized to the year, decade, or [*DATE*] as default value:

  "March 12, 1994" → [*1994*] or
  [*date in the 1990s*]
  "the following day" → [*DATE*].

Heuristics were chosen for these types of entities since they are usually not part of knowledge graphs that can be used to create ontologies.

3.2 Ontology-based generation

Entities of type LOC, ORG, DEM and MISC are generalized using an ontology. The ontology was constructed using Wikidata[2], a knowledge graph where pieces of information are linked together by *properties*. We consider specific membership properties, namely *instance_of* (P31), *subclass_of* (P279), *part_of* (P361), and *is_metaclass_for* (P8225), which express a hierarchical relation from specific to more general, as seen in Table 1.

ID	Label	Example
P31	instance_of	Oslo → capital city
P279	subclass_of	capital city → city
P8225	is_metaclass_for	genre → creative work
P361	part_of	door → house

Table 1: Wikidata properties employed to construct the generalization ontology.

The ontology contains all terms related to humans and their generalizations extracted using the properties mentioned above, with the addition of terms for *countries* and *nationalities*.

3.2.1 Entity linking

The text span to generalize must first be linked to an appropriate term in the ontology. We first search for exact matches, followed by a *contained_in* search in the ontology. Finally, if no entity is found, approximate string matching is employed to tentatively match the PII.[3].

If the above entity linking fails (which means that this term is absent from the ontology), we query Wikidata directly to get a match. If a match is found then we return the results, otherwise '***' is suggested as an appropriate replacement. Masking the term with '***' is both proposed when no match is found and as a final option for all PII, to provide an alternative when the provided generalization options are inappropriate.

3.2.2 Ontology traversal

Every term related to a human in the ontology was used to fetch generalization options using the membership properties in Table 1. In the case of several available property options, the first one is selected. The ordering of properties added to the ontology was: P31, P279, P8225 and finally P361. Below is a list showing the generalizations of the term 'drummer' based on the *P31* property, and of the term 'mother' based on the *P279* property.

drummer → [percussionist] → [instrumentalist] → [musician] → [artist] → [creator] → [person] → ***

mother → [parent] → [first-degree relative] → [kin] → [person] → ***

The generalizations range from the most specific (most informative, but also potentially most risky in terms of identity disclosure) to the less specific (least risky, but also least informative).

4 Dataset

The dataset used for the data collection consists of 553 Wikipedia articles already annotated for text sanitization by Papadopoulou et al. (2022a). Wikipedia articles are suitable for this task as they are both dense in PII and publicly available. For each article, human annotators labelled the text spans that needed to be masked to protect the identity of the mentioned individual, while also retaining as much of the utility of the resulting text as possible. Each text span is also assigned to one of the 8 categories enumerated in Section 3.

[2]See https://www.wikidata.org. The dump file was downloaded on Sept. 13, 2022.

[3]A term is considered a match if the character-level edit distance is below a given threshold, set in our implementation to 15% of the length of the entity string.

Type	Level 1	Level 2	L. > 2	***
DATETIME	1025	1032	360	764
DEM	265	202	242	318
LOC	356	419	263	524
MISC	272	622	481	964
ORG	652	773	430	1066
PERSON	2478	85	0	18
QUANTITY	381	5	0	5
Total	5429	3138	1776	3726

Table 2: Levels of generalization per semantic type.

4.1 Annotation

We expanded the above dataset with the generalization options proposed by the system, and then recruited 9 annotators to select the most suitable replacement among the possible alternatives. To this end, we developed a web based annotation tool through which the annotators received a link to a web page containing the documents they were assigned to annotate. The annotators were also provided with annotation guidelines (see Appendix A 5). The annotators were required to select exactly one option per marked text span. Each annotator annotated 81 documents, whereof 59 were randomly selected, with the remaining 22 documents being multi-annotated. Two examples of text before and after the annotation process is shown below:

Example 1

Original: **Joey Muha** is a Canadian **drummer** from **Port Dover**, Ontario.

Generalized: **[PERSON 1]** is a Canadian **[musician]** from **[town]**, Ontario.

Example 2

Original: **Joakim Lindner** (born **22 March 1991**) is a Swedish footballer who plays for **Varbergs BoIS** as a midfielder. He is son to the competitive sailor **Magnus Olsson**.

Generalized: **[PERSON 1]** (born **[date in the 1990s]**) is a Swedish footballer who plays for **[association football club]** as a midfielder. He is son to the competitive sailor **[PERSON 2]**.

4.2 Analysis

Table 2 details the level of generalization selected by the human annotators according to the entity type. Overall, only 36% of the selections landed on the default '***', meaning that a majority of text spans could be mapped to more meaningful replacement options.

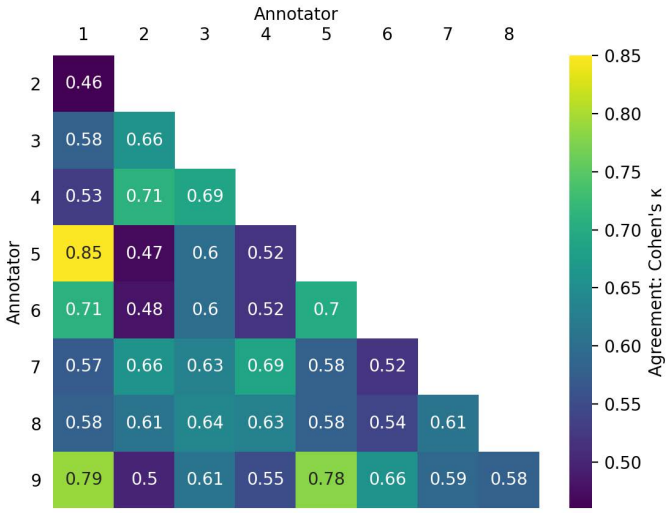


Figure 2: Pairwise agreement between annotators.

The generalization options were sorted from most specific to the most general following the hierarchical structure in Wikidata. Table 2 shows a clear preference for first level generalizations, meaning the annotators selected the least general option more frequently. It should be noted that some semantic categories (PERSON, QUANTITY) had fewer options. For instance, after manual inspection, we observe that 98 % and 96% of all selections made for QUANTITY and PERSON respectively are the least general option. For the MISC category, the corresponding percentage is only 20%.

A subset of the documents were annotated by multiple annotators. We estimated the inter annotator agreement using Light's kappa (L-kappa), as it allows annotators to select from a set of alternatives. It is computed as the mean of the Cohen's kappa of each annotation pair (Conger, 1980). A score of -1 indicates a direct disagreement, while 1 suggests perfect agreement. The L-kappa score obtained this data collection is 0.61, indicating a moderate to substantial agreement. Variations in the agreement between annotator pairs range from 0.46 to 0.85, as shown in Figure 2.

5 Conclusion

We presented an approach to generate replacements for detected PII based on heuristic rules and an ontology derived from Wikidata properties. The approach is employed to enrich an existing text sanitization dataset with suitable replacements for each text span. Those replacements were manually selected by annotators among a set of alternatives generated by the above approach.

The collected data highlights the benefits of this replacement strategy, with 64% of the text spans being mapped to a generalization other than the default '***'. However, the moderate inter-annotator-agreement also illustrates the difficulty of the task, which may admit multiple solutions.

Future work will focus on enriching the ontology, resolving entity linking ambiguities and using the dataset to train a neural model to select appropriate generalizations for PII spans.

Acknowledgments

We acknowledge support from the Norwegian Research Council through the CLEANUP project (grant nr. 308904).

References

Balamurugan Anandan, Chris Clifton, Wei Jiang, Mummoorthy Murugesan, Pedro Pastrana-Camacho, and Luo Si. 2012. T-plausibility: Generalizing words to desensitize text. *Trans. Data Privacy*, 5(3):505–534.

Nikolay Arefyev, Boris Sheludko, Alexander Podolskiy, and Alexander Panchenko. 2020. A comparative study of lexical substitution approaches based on neural language models. *arXiv preprint arXiv:2006.00031*.

Synnøve Bråthen, Wilhelm Wie, and Hercules Dalianis. 2021. Creating and evaluating a synthetic Norwegian clinical corpus for de-identification. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 222–230, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

David Carrell, Bradley Malin, John Aberdeen, Samuel Bayer, Cheryl Clark, Ben Wellner, and Lynette Hirschman. 2012. Hiding in plain sight: use of realistic surrogates to reduce exposure of protected health information in clinical text. *Journal of the American Medical Informatics Association*, 20(2):342–348.

Anthony J. Conger. 1980. Integration and generalization of kappas for multiple raters. *Psychological Bulletin*, 88:322–328.

Hercules Dalianis. 2019. Pseudonymisation of Swedish electronic patient records using a rule-based approach. In *Proceedings of the Workshop on NLP and Pseudonymisation*, pages 16–23, Turku, Finland. Linköping Electronic Press.

Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. 2017. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606.

Mark Elliot, Elaine Mackey, Kieron O'Hara, and Caroline Tudor. 2016. *The Anonymisation Decision Making Framework*. UK Anonymisation Network, United Kingdom.

GDPR. 2016. General Data Protection Regulation. European Union Regulation 2016/679.

Pierre Lison, Ildikó Pilán, David Sanchez, Montserrat Batet, and Lilja Øvrelid. 2021. Anonymisation models for text data: State of the art, challenges and future directions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4188–4203, Online. Association for Computational Linguistics.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic. Association for Computational Linguistics.

Anthi Papadopoulou, Pierre Lison, Lilja Øvrelid, and Ildikó Pilán. 2022a. Bootstrapping text anonymization models with distant supervision. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4477–4487, Marseille, France. European Language Resources Association.

Anthi Papadopoulou, Yunhao Yu, Pierre Lison, and Lilja Øvrelid. 2022b. Neural text sanitization with explicit measures of privacy risk. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 217–229, Online only. Association for Computational Linguistics.

Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. 2022. The Text Anonymization Benchmark (TAB): A Dedicated Corpus and Evaluation Framework for Text Anonymization. *Computational Linguistics*, 48(4):1053–1101.

David Sánchez and Montserrat Batet. 2016. C-sanitized: A privacy model for document redaction and sanitization. *Journal of the Association for Information Science and Technology*, 67(1):148–163.

Elena Volodina, Yousuf Ali Mohammed, Sandra Derbring, Arild Matsson, and Beata Megyesi. 2020. Towards privacy by design in learner corpora research: A case of on-the-fly pseudonymization of Swedish learner essays. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 357–369, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Emily M Weitzenboeck, Pierre Lison, Malgorzata Cyndecka, and Malcolm Langford. 2022. The GDPR

and unstructured data: is anonymization possible? *International Data Privacy Law*, 12(3):184–206.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. BERT-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence, Italy. Association for Computational Linguistics.

Appendix A. Annotation guidelines

The annotation guidelines describing the task, along with examples, are presented below.

Replacement Choices in Text Sanitization: Annotation Guidelines

This annotation effort is part of a larger research project that seeks to understand how to automatically remove personally identifiable information from text documents (a problem called *text sanitization*). Personally identifiable information refers to any piece of information that may directly or indirectly reveal the identity of a particular individual. Text sanitization is an important problem when dealing with sensitive documents where we need to conceal the identity of given person(s) to protect their privacy.

The result of your annotation work will be included in a new, public dataset released under an open-source license.

The Task

In this task, you are given a number of short biographies extracted from Wikipedia. To conceal the identity of the individual described in the biography, some text spans have already been marked as needing to be replaced. Each text span is shown in a drop-down menu where the values correspond to possible replacements. The original text span for which you will choose a replacement is also provided to help in the decision making process.

Your job is to select in each drop-down menu the best replacement for the text span according to the following two criteria:

1. The replacement should not disclose (directly or indirectly) the person's identity.
2. Provided that the above criteria is satisfied, the replacement should be as informative and readable as possible.

For example, in the sentence:

PERSON 1 lives and works in ***Oslo*** *...*

possible choices for ***'Oslo'*** might include *[capital of Norway]*, *[city in Norway]*, *[city]* and *'***'*. The first choice is not general enough since it is as informative as the word Oslo. The second choice is more general, followed by the third choice, and finally the default *'***'*, which is least informative (but also least risky from a privacy perspective). Person names are by default replaced by *PERSON X* (where X is an integer).

Procedure

The annotation work consists of the following steps:

- **Step 1** Read through the text once.
- **Step 2** For each marked span, look at the list of possible replacements and pick the most appropriate one. Only one replacement can be selected for each text span.
- **Step 3** When you are done with all replacements, review the text one final time. The selected replacements should not disclose the person identity, and the text should be as informative and readable and possible.

Many suggested replacements will be incorrect or irrelevant – this is normal and expected. If none of the suggested replacements are suitable for a given text span, you should choose the default '***' option.

The '***' option

In all the dropout lists of possible replacements, there will be an *'***'* option. Use this if you find that no other replacement is appropriate.

Sometimes the *'***'* is the only suitable option, since you might encounter cases where the automatic generation of suggested replacements failed to come up with good options.

Corner cases

There might be cases where a replacement looks appropriate but does not entirely fit the form of the sentence. For example, in the following sentence:

PERSON 1 was born on ***May 18, 1943*** *[...]*

The possible replacements will be *[1943]*, *[date in the 1940s]* and *'***'*. The most suitable choices in this case are *[1943]* and *[date in the 1940s]* (although it might necessitate some rephrasing to fit the current form of the sentence), not *'***'*.

Example

Below you will find a step-by-step example of the annotation steps.

Start by briefly reading the text **(Step 1)**

Then for each of the spans choose one replacement **(Step 2)**. Following is a possible set of replacements chosen.

For example, the two decades could be replaced with the *'***'* option since they provide additional information along with the rest of the personal information still left in the text (e.g. *British, gay rights activist, general secretary* etc.) that could lead to the person being identified easier, which we wish to prevent.

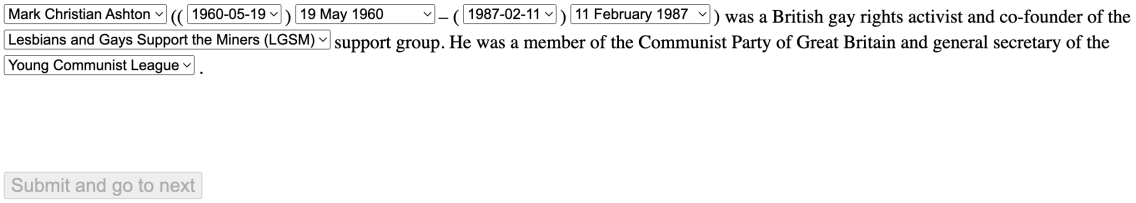


Figure 1: Step 1

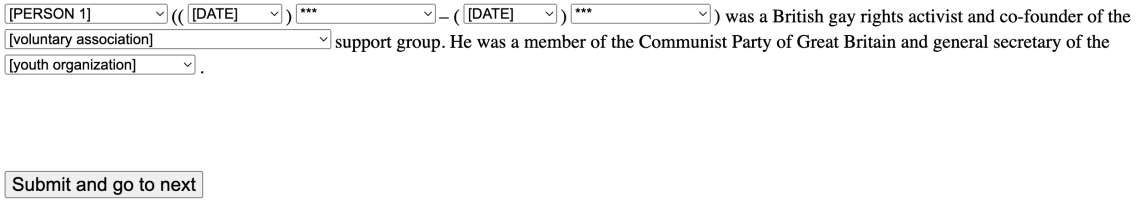


Figure 2: Step 2

Note that there is no one correct solution, as long as the identity of the individual is not disclosed and the replacement choices result in an (as much as possible) informative text.

NB!You have to choose a replacement option. The original string is provided (first option in the drop-down list that cannot be chosen) in order to help choose the most appropriate one. The *Submit and go to next* button can only be clicked if replacements for all the spans have been selected.

Read the text with the selected replacements one last time **(Step 3)**. Make sure that you have chosen replacements for all text spans. Click on *Submit and go to next* to continue with the rest of the texts for this task.

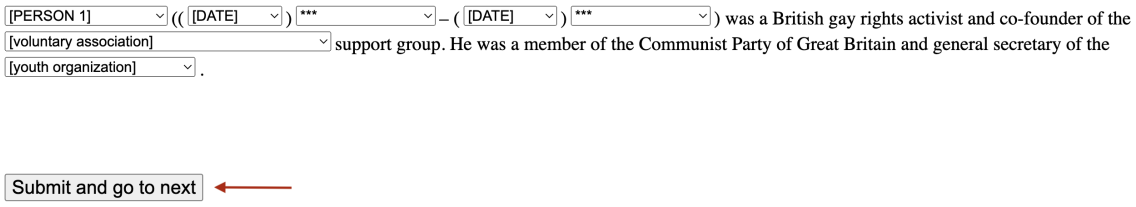


Figure 3: Step 3

A short message will appear on your screen when your assigned number of texts have been annotated.

MeDa-BERT: A medical Danish pretrained transformer model

Jannik Skyttegaard Pedersen*
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark
jasp@mmmi.sdu.dk

Martin Sundahl Laursen*
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark
msla@mmmi.sdu.dk

Pernille Just Vinholt
Department of Clinical Biochemistry
Odense University Hospital

Thiusius Rajeeth Savarimuthu
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark

Abstract

This paper introduces a medical Danish BERT-based language model (MeDa-BERT) and medical Danish word embeddings. The word embeddings and MeDa-BERT were pretrained on a new medical Danish corpus consisting of 133M tokens from medical Danish books and text from the internet. The models showed improved performance over general-domain models on medical Danish classification tasks. The medical word embeddings[1] and MeDa-BERT[2] are publicly available.

1 Introductions

Large language models (LLM) are powerful representation learners and have become the backbone structure of many modern natural language processing (NLP) systems. To learn text representations, LLM are first pretrained on a large-scale text corpus using self-supervised learning, e.g., masked language modelling. After pretraining, LLM are fine-tuned on specific downstream tasks where they have achieved state-of-the-art results on NLP benchmarks such as GLUE (Wang et al., 2018).

However, directly applying these general pretrained models to specialized domains such as the medical have led to unsatisfactory results (Peng et al., 2019). As a solution to this, a second round of in-domain pretraining (domain-adaptive pretraining) has shown to improve the performance of LLMs that were first trained on a general domain corpus (Gururangan et al., 2020). Domain-adaptive pretraining adjusts the weights of the LLM to better capture the terminology, style, and nuances that are relevant to the target domain.

Resource-rich languages such as English have large domain-specific corpuses available that have been used to develop e.g., biomedical (Lee et al., 2020), clinical (Alsentzer et al., 2019), scientific (Beltagy et al., 2019), and financial (Peng et al., 2021) LLMs that perform better than models trained on general corpuses. These models could potentially be used to improve human decision making, save time, and reduce costs, e.g., by extracting information from scientific articles, identifying potential drug interactions, and helping with NLP tasks such as text classification, named entity recognition, and question answering for each of their specialized domains.

For the Danish language, only LLMs trained on a general domain have been made publicly available[3]. This paper presents a medical Danish BERT model (MeDa-BERT)—a LLM trained on a new medical Danish text corpus. We also used the medical corpus to train medical word embeddings as they still have value in the clinical domain (Laursen et al., 2023). To evaluate the medical word embeddings and MeDa-BERT, we used existing medical Danish classification datasets. We found that an LSTM model using the medical word embeddings outperformed a similar model using general-domain word embeddings, and that MeDa-BERT performed slightly better than a general-domain BERT model.

2 Method

This section first describes how the medical corpus was collected and used to pretrain the medical Danish word embeddings and MeDa-BERT. Next, the datasets used to compare model performances and the fine-tuning procedure is described.

*Equal contribution

[1] https://huggingface.co/jannikskytt/MeDa-WE

[2] https://huggingface.co/jannikskytt/MeDa-Bert

[3] Pedersen et al. (2022b) developed a clinical transformer model but it is not publicly available

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 301–307

Corpus	Type	Date retrieved	Tokens
Clinical guidelines	Guidelines	October - November 2022	80,567,576
Medicin.dk	Information portal	June 2021	28,878,335
FADL	Books	January 2022	12,531,373
Sundhed.dk	Information portal	May 2022	6,767,409
Netdoktor.dk	Information portal	October 2022	3,227,051
Wikipedia	Encyclopedia	October 2022	1,992,796
Total			**133,964,540**

Table 1: Number of tokens and date retrieved for each data source

2.1 Danish medical corpus

We collected data from the internet and from medical books. The owners of the data resources approved that we used their data in this study. We describe the data collection for each text contributor below. An overview of the text corpuses and their size can be seen in Table 1.

2.1.1 Clinical guidelines

We collected text from the document management systems of the five Danish regions. The documents contain guidelines and instructions for diagnostics and treatment of patients and all workflows that support this. The document systems also include non-medical documents from e.g. purchasing, logistics, and service departments which were removed. All departments that were excluded and the number of tokens retrieved from each region can be seen in Appendix A.

2.1.2 Medical information portals

We collected text from webpages that provide information to medical doctors and patients. The text was collected from Medicin.dk, Netdoktor.dk, and Sundhed.dk. The resources provide information about diseases, symptoms, and medical treatments. Moreover, the resources contain information specifically for health care professionals, e.g., medication guidelines and information about best practices in the field. Text not related to the medical domain and text written by non-professionals were removed from the corpus. A description of this process can be seen in appendix A.

2.1.3 Books

This part of the corpus consisted of 107 medical books from publisher FADLs Forlag that publishes books for medicine and nursing school.

2.1.4 Wikipedia

We used PetScan[4] to search for medical Wikipedia documents within predefined categories and its subcategories. We used a maximum depth of 5 for searching for subcategories. The following categories were used: anatomi, physiology, diseases, medication, epidemiology, diagnostics, medical procedures, medical specialities, medical physics, and medical equipment. We excluded documents with the categories: persons and companies. This process resulted in 5,391 documents. Next, we manually removed non-medical articles from that list which resulted in 5,266 documents.

2.2 Preprocessing of data

For all text corpusses, we defined a sample as one paragraph, i.e., a continuous stream of text without line breaks. We inserted spaces between alphanumeric and non-alphanumeric characters. Samples were further preprocessed to fit the pretraining procedure for either word embeddings or the transformer model, as detailed below.

2.2.1 Danish medical transformer model

MeDa-BERT was initialized with weights from a pretrained Danish BERT model[5] trained on 10.7 GB Danish text from Common Crawl (9.5 GB), Danish Wikipedia (221 MB), debate forums (168 MB), and Danish OpenSubtitles (881 MB).

For domain-adaptive pretraining, samples from the collected medical corpus were appended a [CLS] and [SEP] token in the start and end of each sample, respectively. Samples were concatenated to fit the maximum sequence length of 512 tokens and document boundaries were indicated by adding an extra [SEP] token in between samples. After this process, we removed duplicates corresponding to 0.2% of the total corpus. The model was trained using Adam (Kingma and Ba, 2015) with a weight decay of 0.01 as described in (Loshchilov and Hutter). Using gradient accumulation, the model was trained with a batch size of 4,032, a learning rate of 1e-4, and a linear learning rate decay warmed up over 1 epoch. The model was pretrained for a total of 48 epochs and evaluated after 16, 32, and 48 epochs. We used 5% of the samples as a validation set to evaluate the model during pretraining and trained the model on the remaining data using dynamic

[4] https://petscan.wmflabs.org/

[5] https://github.com/certainlyio/nordic_bert

Dataset	Label	Train	Validation	Test
Bleeding	Positive	10,331	1,300	1,300
	Negative	10,331	1,300	1,300
Bleeding site	Airways	1,000	125	125
	Cerebral	1,000	125	125
	Ear-nose-throat	1,000	125	125
	Eyes	1,000	125	125
	Gastrointestinal	1,000	125	125
	Gynecological	1,000	125	125
	Internal	1,000	125	125
	Skin	1,000	125	125
	Urogenital	1,000	125	125
	Unknown	1,000	125	125
VTE	Positive	9,064	1,100	1,100
	Negative	9,064	1,100	1,100
VTE site	Airways	1,600	200	200
	Lungs	1,600	200	200
	Unknown	1,600	200	200

Table 2: Dataset distributions

masked language modeling. The model was optimized using four Tesla v100 GPUs using the Huggingface (Wolf et al., 2020) library. All model parameters and pretraining losses are shown in Appendix B.

2.3 Danish medical word embeddings

We trained 300-dimensional FastText (Bojanowski et al., 2017) word embeddings. The embeddings were trained for 10 epochs using a window size of 5 and 10 negative samples. The hyperparameters were chosen to be able to compare the produced embeddings with the Danish FastText word embeddings from Grave et al. (2018) that were trained on a general domain.

2.4 Datasets

We compared performances between models using four medical datasets: bleeding classification, bleeding site classification, venous thromboembolism (VTE) classification, and VTE site classification. All samples were annotated with a consensus label from three medical doctors. The dataset distributions can be seen in Table 2 and examples of samples can be seen in Appendix C.

2.4.1 Bleeding classification

The bleeding dataset (Pedersen et al., 2021) is a binary classification problem with 25,862 samples. The dataset was constructed from 900 Danish electronic health records (EHR) from Odense University Hospital. The samples had an average token length of 13.3.

2.4.2 Bleeding site classification

The bleeding site dataset (Pedersen et al., 2022b) is a 10-class classification problem with 11,250 unique bleeding-positive samples annotated for the bleeding site. The bleeding site labels were: airways, cerebral, ear-nose-throat, eyes, gastrointestinal, gynecological, internal, skin, urogenital, and unknown. The dataset was constructed from 149,523 Danish EHR notes from Odense University Hospital. The samples had an average token length of 14.4.

2.4.3 VTE classification

The VTE dataset (Pedersen et al., 2022a) is a binary classification problem with 22,528 samples. The dataset was constructed from 94,520 Danish EHR notes from Odense University hospital. The samples had an average token length of 13.8.

2.4.4 VTE site classification

The VTE site dataset (Pedersen et al., 2022a) is a 3-class classification problem with 6,000 VTE-positive samples annotated for the VTE site. The VTE site labels were: airways, lungs, and unknown. The dataset was constructed from 94,520 Danish EHR notes from Odense University Hospital. The samples had an average token length of 14.5.

2.5 Fine-tuning

2.5.1 MeDa-BERT and BERT

We used the [CLS] token followed by a classification layer to classify samples of the datasets. We searched for the best models five times using Adam with learning rates [5e-5, 3e-5, 1e-5], i.e., we fine-tuned each model 15 times. The models were trained for a maximum of 10 epochs.

2.5.2 LSTM

We used the medical word embeddings as input to a bidirectional LSTM layer with a hidden layer size of 512. The last hidden state of the LSTM was followed by a dropout layer with probability 0.2, a dense layer of size 256, a ReLU activation function, a dropout layer of probability 0.2, and a dense classification layer. This model is referred to as LSTM+MeDa-WE.

The performance of the model is compared with another LSTM model (LSTM+General-WE) with the same parameters but using FastText embeddings trained on the general domain as input (Grave et al., 2018). We searched for the best

	Bleeding	Bleeding site	VTE	VTE site
LSTM+General-WE	$83.8_{.7}$	$69.3_{.8}$	$88.5_{.2}$	$86.4_{.7}$
LSTM+MeDa-WE	$\mathbf{91.4_{.3}}$	$\mathbf{84.9_{1.1}}$	$\mathbf{94.1_{.3}}$	$\mathbf{93.4_{.4}}$
BERT	$94.3_{.6}$	$86.7_{.8}$	$96.7_{.3}$	$94.7_{.3}$
MeDa-BERT_16	$94.7_{.3}$	$88.4_{.6}$	$\mathbf{97.1_{.4}}$	$95.5_{.2}$
MeDa-BERT_32	$95.1_{.5}$	$88.7_{.6}$	$96.9_{.3}$	$95.7_{.3}$
MeDa-BERT_48	$\mathbf{95.3_{.4}}$	$\mathbf{89.1_{.2}}$	$97.0_{.5}$	$\mathbf{95.8_{.3}}$

Table 3: Mean accuracy and standard deviation (subscript) for each model on four medical classification tasks. Best results for the LSTM and BERT-based models highlighted in bold. MeDa-BERT_16 denotes the MeDa-BERT model pretrained for 16 epochs.

models five times using Adam with learning rates [5e-5, 3e-5, 1e-5], i.e., we fine-tuned each model 15 times.

For all models we report the mean test set accuracy and standard deviation for the five best performing models on the validation dataset.

3 Results

Table 3 shows the results of each model on the four classification datasets.

3.1 Word embedding comparison

Using the medical word embeddings as input to an LSTM model resulted in large improvements compared to using general word embeddings. On average, LSTM+MeDa-WE outperformed the LSTM+General-WE model by 8.9 percentage points (PP). The largest improvement was seen on the 10-class bleeding site classification with an improvement of 15.6 PP.

3.2 Language model comparison

Comparing BERT and MeDa-BERT, the performance improvements were smaller. MeDa-BERT performed better on three of the datasets with an average improvement of 1.2 PP. The largest improvement was on the 10-class bleeding site classification with an improvement of 2.4 PP.

4 Discussion and limitations

This paper presented a new Danish medical corpus that was used to train NLP models. The corpus included medical books and text scraped from medical websites that provide information for both citizens and healthcare professionals. We applied different techniques to filter out non-medical data, e.g., by removing documents from non-medical departments or text written by non-healthcare professionals. While these steps did remove a large part of non-medical text, some non-medical text might still be present in the corpus. However, the results showed that models pretrained on the medical corpus performed better than general-domain models, especially for multiclass classification problems.

For the Danish language, few medical evaluation datasets are available and therefore the models were only evaluated on classification tasks. Moreover, the evaluation datasets were constructed from EHR text which has its own nuances compared to the text of the medical pretraining corpus, e.g., EHR text contains many spelling mistakes whereas the medical corpus contains few grammatical errors. These factors might limit the generalizability of the results. Future work should evaluate the models on other tasks, e.g., named-entity recognition and question answering which will provide a better understanding of the models' capabilities.

We found continuous small performance improvements by pretraining MeDa-BERT for more epochs. The model might improve with further pretraining but because of limited computational resources and the small rate of improvement, we did not explore this further. The model would also benefit from more medical pretraining data. Although this paper presented a large part of the available medical Danish text, more data could be collected, e.g., from other medical book publishers and websites.

The medical datasets used to evaluate the models are not publicly available because of privacy concerns. For future work, we will strive to publish parts of the medical corpus which requires permission from the text owners. We advise interested researchers to contact us for sharing possibilities.

5 Conclusion

This paper presented a Danish medical corpus consisting of 133M tokens. The corpus was used to pretrain medical word embeddings and language models. The models trained on the medical corpus performed better than similar models trained on a general domain.

Acknowledgement

We would like to thank the medical corpus contributors who gave acceptance of using their text in this project. Listed in alphabetical order: FADL's forlag, Medicin.dk, Netdoktor.dk, Sundhed.dk, The Capital Region of Denmark, The Central Region of Denmark, The Region of Northern Denmark, The Region of Southern Denmark, The Region of Zealand.

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. https://doi.org/10.18653/v1/W19-1909 Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. https://doi.org/10.18653/v1/D19-1371 SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. https://aclanthology.org/Q17-1010 Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5.

Édouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomáš Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. https://doi.org/10.18653/v1/2020.acl-main.740 Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. http://arxiv.org/abs/1412.6980 Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Martin Sundahl Laursen, Jannik Skyttegaard Pedersen, Pernille Just Vinholt, Rasmus Søgaard Hansen, and Thiusius Rajeeth Savarimuthu. 2023. Benchmark for evaluation of danish clinical word embeddings. *Northern European Journal of Language Technology*, 9(1).

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Jannik Pedersen, Martin Laursen, Pernille Just, Anne Alnor, and Thiusius Savarimuthu. 2022a. Investigating anatomical bias in clinical machine learning algorithms. In *Findings of the European Chapter of the Association for Computational Linguistics: EACL 2023*, Dubrovnik, Croatia. Association for Computational Linguistics.

Jannik S Pedersen, Martin S Laursen, Thiusius Rajeeth Savarimuthu, Rasmus Søgaard Hansen, Anne Bryde Alnor, Kristian Voss Bjerre, Ina Mathilde Kjær, Charlotte Gils, Anne-Sofie Faarvang Thorsen, Eline Sandvig Andersen, et al. 2021. Deep learning detects and visualizes bleeding events in electronic health records. *Research and practice in thrombosis and haemostasis*, 5(4):e12505.

Jannik S Pedersen, Martin S Laursen, Cristina Soguero-Ruiz, Thiusius R Savarimuthu, Rasmus Søgaard Hansen, and Pernille J Vinholt. 2022b. Domain over size: Clinical electra surpasses general bert for bleeding site classification in the free text of electronic health records. In *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 1–4. IEEE.

Bo Peng, Emmanuele Chersoni, Yu-Yin Hsu, and Chu-Ren Huang. 2021. Is domain adaptation worth your investment? comparing bert and finbert on financial tasks. In *Proceedings of the Third Workshop on Economics and Natural Language Processing*, pages 37–44.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. https://doi.org/10.18653/v1/W19-5006 Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. https://doi.org/10.18653/v1/W18-5446 GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing*

and Interpreting Neural Networks for NLP, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. https://doi.org/10.18653/v1/2020.emnlp-demos.6 Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Appendices

A Preprocessing of text corpuses

A.1 Medical information portals

Netdoktor.dk provides information about diseases, symptoms, medication, and treatment. Netdoktor.dk contains sections that are not related to the medical domain and discussion forums where users can communicate. Therefore, we removed documents having links containing the following strings: debat, kultur, testdigselv, behandlerguiden, nyhedsbrev, nyheder, privacypolicy, kontaktnetdoktor, cookieinformation, disclaimer, sponsorindhold and discussions. Moreover, citizens can ask medical questions [6] that are answered by health care professionals. We only included the answers to these questions.

Medicin.dk has three sub-pages: `www.min.medicin.dk` that provides information to citizens, `www.pro.medicin.dk` that provides information to health care professionals, and `www.indlaegssedler.dk` that contains information about medicine. We included all documents from these webpages.

Sundhed.dk provides information for medical professionals [7] and citizens [8] about diseases, symptoms, medication and treatment. We included all documents from these webpages.

A.2 Clinical guidelines

We collected clinical guidelines from the 5 regions of Denmark: The Capital Region of Denmark, The Region of Northern Denmark, The Region of Southern Denmark, The Region of Zealand, and The Central Region of Denmark. For each region we removed non-medical documents, seen in Table 4.

[6] `https://www.netdoktor.dk/brevkasser/`

[7] `https://www.sundhed.dk/sundhedsfaglig/`

[8] `https://www.sundhed.dk/borger/`

Region	Categories removed (in Danish)	Date retrieved	Tokens
Capital Region	Den sociale virksomhed Center for ejendomme Center for HR Center for Regional Udvikling Region Hovedstadens Apotek Steno Diabetes Center Copenhagena	October 2022	13,443,269
Northern Region	Logistik afdeling Teknisk Afdeling Himmerland Teknik Logistik Service	October 2022	6,505,559
Southern Region	Administration Service PsykInfo	September - November 2022	29,075,187
Region Zealand	Administration HR organisation og ledelse Indkøb IT PortørCentral Rengøring Økonomi Uddannelse	November 2022	6,387,083
Central Region		November 2022	25,156,478

Table 4: Categories removed and number of tokens from each of the Danish regions.

Parameter	Value
Architecture	
Number of layers	12
Hidden size	768
FFN inner hidden size	3072
Attention heads	12
Attention head size	64
Dropout	0.1
Attention dropout	0.1
Max seq. length	512
Optimization	
Learning rate	1e-4
Optimizer	AdamW
Adam weight decay	0.01
Adam epsilon	1e-6
Adam beta1	0.90
Adam beta2	0.98
Learning rate decay	Linear
Batch size	4032
Warm up	1 epoch
Epochs	16, 32, 48
Gradient clipping	1.0

Table 5: Architecture and optimization parameters for pretraining MeDa-BERT

B Model parameters and pretrainng loss

Table 5 shows the architecture and optimization parameters for pretraining MeDa-BERT. Table 6 shows the masked language modelling loss for MeDa-BERT during pretraining.

	Train loss	Validation loss
MeDa-BERT_16	2.122	2.019
MeDa-BERT_32	1.874	1.792
MeDa-BERT_48	1.766	1.673

Table 6: Masked language modelling loss for MeDa-BERT during pretraining. MeDa-BERT_16 denotes the model pretrained for 16 epochs.

C Dataset examples

Figure 7 shows a sample from each dataset translated from Danish to English.

Dataset	Example	Label
Bleeding	"Girl hospitalized on 14.05.11 with bleeding tendency. 1½ years ago, noticed bleeding on both arms, under the armpits and on the inner thighs. Subsequently blood discharges on the mucous membrane of the cheeks and quite heavy menstrual bleeding, which is unusual for pt."	Positive
Bleeding site	"19-year-old man referred by on-call doctor due to sudden onset of macroscopic hematuria and left-sided flank pain."	Urogenital
VTE	"Pt has severe heart failure and hence dyspnoea and the feeling of air hunger, and, in addition, pt has pulmonary embolisms and COPD. Treatment with Fragmin has started."	Positive
VTE site	"Irregular contours on the left side in the transverse sinus and beginning part of the sigmoid sinus compatible with partial thrombosis."	Brain

Table 7: Example of a sample from each dataset translated from Danish to English.

Standardising Pronunciation for a Grapheme-to-Phoneme Converter for Faroese

Sandra Saxov Lamhauge[1], Iben Nyholm Debess[1], Carlos Daniel Hernández Mena[2], Annika Simonsen[3], Jon Gudnason[2]
[1]The University of the Faroe Islands
[2]Reykjavík University
[3]The University of Iceland
sandrasl@setur.fo, ibennd@setur.fo, carlosm@ru.is, annika@hi.is, jg@ru.is

Abstract

Pronunciation dictionaries allow computational modelling of the pronunciation of words in a certain language and are widely used in speech technologies, especially in the fields of speech recognition and synthesis. On the other hand, a grapheme-to-phoneme tool is a generalization of a pronunciation dictionary that is not limited to a given and finite vocabulary. In this paper, we present a set of standardized phonological rules for the Faroese language; we introduce FARSAMPA, a machine-readable character set suitable for phonetic transcription of Faroese, and we present a set of grapheme-to-phoneme models for Faroese, which are publicly available and shared under a creative commons license. We present the G2P converter and evaluate the performance. The evaluation shows reliable results that demonstrate the quality of the data.

1 Introduction

Pronunciation dictionaries are important components in speech technology (i.e. for ASR and TTS). They are used to link orthographic forms with their pronunciations. There are many kinds of pronunciation dictionaries; some that only provide a standard pronunciation (Weide, 1998), and some that also provide dialectal variants (Robinson, BEEP dictionary). However, dictionaries are always going to be limited to their entries; it is therefore that one would train grapheme-to-phoneme (G2P) models on pronunciation dictionaries and use those models to automatically transcribe unknown words (Nikulásdóttir et al., 2018).

In recent years, there has been steady progress in Faroese speech technology, and the work is ongoing (e.g. Ingason et al., 2012; Hernández Mena et al., 2022a; 2022b; 2022c). A Faroese pronunciation dictionary was created for the first Faroese ASR project, Ravnur and published in 2022 as part of a Basic Language Resource Kit (BLARK) for Faroese (Debess et al., 2022; Simonsen et al., 2022)[1]. A BLARK is defined as the minimal collection of language resources needed to develop language technology for a specific language (Krauwer, 2003; Maegaard et al., 2006). However, the BLARK is not limited to ASR, but can be used to develop a wide range of LT for Faroese.

The dictionary from the Ravnur project uses the variation spoken in the capital of the Faroe Islands as a standard (an arbitrary choice as discussed in Section 5). When creating the grapheme-to-phoneme (G2P) models, it was decided to expand the dictionary into two different pronunciation dictionaries based on dialectal difference, one CENTRAL and one EAST. This way, we could cover more ground and open up the possibility to expand the work further in the future.

Because this was the first open source Faroese dictionary of its kind, Ravnur also had to create a Faroese SAMPA alphabet (called FARSAMPA here) to use for the pronunciations. The process and decisions made when creating FARSAMPA

[1] The BLARK for Faroese is open source, published under a CC BY 4.0 licence on the platform OpenSLR (Debess et al., 2022).

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 308–317

and the Faroese pronunciation dictionary have never been published until now. We therefore set out to describe the work that went behind creating these fundamental language resources and how we adapted them to create a reliable G2P model.

In Section 2, we present FARSAMPA. Section 3 outlines some of the most common grapheme-to-phoneme conversions of Faroese, while Section 4 presents the standardization of phonetic variation for the G2P. In Section 5, we discuss Faroese dialectal variation and the decision to make two different pronunciation dictionaries to train G2P models on. Finally, we introduce our G2P models that have been trained on the two pronunciation dictionaries, present an evaluation, and conclude.

Consonants	FARSAMPA	IPA	Vowels	FARSAMPA	IPA
Stops	p	pʰ	Monophthongs	i	i
	b	p		I	ɪ
	t	tʰ		e	e
	d	t		E	ɛ
	k	kʰ		a	a
	g	k		y	y
Fricatives	f	f		Y	ʏ
	v	v		2	ø
	4	ð		9	œ
	5	θ		u	u
	s	s		U	ʊ
	S	ʃ		o	o
	z	ʂ		O	ɔ
	h	h		8	ə
Affricates	tS	tʃʰ	Diphthongs	EA	ɛa
	dZ	tʃ		OA	ɔa
Nasals	m	m		UJ	ʊi
	M	m̥		EJ	ɛi
	n	n		aJ	ai
	x	n̥		aW	au
	N	ŋ		OJ	ɔi
	X	ŋ̊		OW	ɔu
Laterals	l	l		3W	ʉu
	L	l̥		EW	ɛu
Approximants	w	w		9W	œu
	j	j		9J	œi
	r	ɹ	Diacritics	H	ʰ

Table **1**: Overview of FARSAMPA mapped to IPA-symbols[2].

2 FARSAMPA – Faroese SAMPA[3]

This section introduces FARSAMPA: a machine-readable character set suitable for phonetic transcription of Faroese. This character set is a phonetic alphabet consisting of an inventory of ASCII symbols mapped onto symbols of IPA (the International Phonetic Alphabet). The inventory content is based on Faroese phonetic and phonological knowledge and research (e.g. Petersen, 2021; Thráinsson et al., 2012), and transcription conventions for Faroese were considered when creating the alphabet. All phonemes in the G2P tool introduced in this article are written in FARSAMPA, as the lexical data of the tool is transcribed in FARSAMPA. See Table 1 for an overview of the inventory and characters in FARSAMPA. FARSAMPA was developed as a part of the BLARK in the Ravnur project. The alphabet covers all Faroese phonemes and a few allophones. The alphabet was used for transcribing all words in the Ravnur lexicon (350.000 word forms) and has thereby been tested, adjusted, and proved suitable and sufficient for Faroese phonetic transcription when working with language technology. Every character in the alphabet is directly translatable to an IPA character, making all transcriptions readily convertible to IPA or other systems mapped up against IPA.

The primary purpose of FARSAMPA is to make phonetic transcriptions machine-readable and ready for automatic processing, and even though the value of characters is arbitrary to a machine, we need to keep in mind that phonetic transcription also entails a manual element and needs to be somewhat human-readable as well.

The designation of characters to phones was based on simplicity, efficiency, and intuition. The basic guidelines[4] are summed up as follows:

- All IPA symbols that coincide with a lowercase letter from the latin alphabet are designated the same character in FARSAMPA (e.g. ‘p’, ‘f’, ‘o’).

[2] Note that the FARSAMPA also includes length distinction, primary and secondary stress (:, %, ~), but these suprasegmental attributes have not been used in the phonetic transcriptions of the words in the data set for this project making the G2P-tool and will therefore not be introduced further at this point.

[3] The use of the term SAMPA stems from this project: https://www.phon.ucl.ac.uk/home/sampa/. The Faroese SAMPA from the Ravnur project is not originally published under the name FARSAMPA, but we will use FARSAMPA to refer to the Faroese SAMPA in this article.

[4] Based on the original SAMPA recommendations for Danish, Dutch, English, French, German and Italian, https://www.phon.ucl.ac.uk/home/sampa/.

- Categorical similarities, e.g. ‘O / ɔ’ and ‘N / ŋ’ (based on standard SAMPA guidelines).
- Graphic similarities, e.g., ‘S / ʃ’ and ‘U / ʊ‘.
- With no letter available, numbers are used over other keyboard symbols as they are easily accessible on qwerty-keyboards language wide and easier to name.
- Avoid designating counterintuitive characters, e.g. ‘T’ for a vowel. In those cases, numbers or other symbols are preferable.

Not all allophones are represented, as that level of detail does not benefit accuracy or efficiency of language technology purposes. Four unvoiced allophones of the sonorant phonemes /m, n, N, l/ are included, as they are acoustically very different from their voiced counterparts. The only diacritic included with its own character is the preaspiration /H/. Postaspiration and other relevant diacritics for Faroese (e.g. voiced/unvoiced) are integrated in the character system and not denoted via symbols of their own (e.g. [p] and [b] only differ by postaspiration, [m] and [M] only differ by voicing).

3 Phonological rules of Faroese

In this section, we will outline some of the most common grapheme-to-phoneme conversions of (central) Faroese. For dialectal variation, see Section 5. The conversions are based on Thráinsson et al. (2012). Almost all of the rules listed have exceptions, and there are other phonological rules as well. For a more thorough overview, see Thráinsson et al. (2012). Furthermore, the phonetics of Faroese are generally understudied (Lamhauge, 2022), and therefore, many phonological conditions have not been sufficiently studied or described. Some of these phonological conditions will be discussed as well.

3.1 Grapheme-to-phoneme conversions

In native Faroese words, stressed vowels are short when followed by two or more consonants, except after the following consonant clusters*: pr, pl, tj, tr, kj, kr, kl, sj*[5]. Stressed vowels are long in all other positions. Table 3 (next page) shows the different graphemes representing vowels in Faroese and their phonemic counterparts.

Table 2 shows the most frequent pronunciation of the consonants in Faroese. In certain grapheme combinations, the consonants have different pronunciations. Some of these are listed in Table 4.

Grapheme combinations	**Phoneme**	**Example**
b, bb	[b], [b:]	bátur ‘boat’, abbi ‘grandfater’
d, dd	[d], [d:]	dust ‘dust’, koddi ‘pillow’
g, gg	[g], [g:]	gala ‘to crow’, sjagga ‘to twaddle’
p, pp	[p], [Hb:]	pílur ‘arrow’, mappa ‘folder’
t	[t], [Hd:]	tekja ‘roof’, detta ‘fall’
k	[k], [Hg:]	kaka ‘cake’, krakkur ‘stool'
f	[f], [f:]	fara ‘to go’, skaffa ‘to provide’
v	[v]	øvund ‘envy’
n, nn	[n], [n:]	nú ‘now’, kanna ‘jug’
n before g or k	[N], [X]	ganga ‘to walk’, banka ‘to knock’
m, mm	[m], [m:]	koma ‘come’, ramma ‘frame’
l	[l]	ala ‘breed’
r	[r][6]	læra ‘to learn’, marra‘nightmare’
h	[h]	hús ‘house’
j	[j]	ja ‘yes’
s, ss	[s], [ss]	siga ‘to say’, kassi ‘box’

Table **2**: The most frequent pronunciation of consonants in Faroese.

Palatalization occurs in Faroese of the phonemes /g, k/ in front of the front, unrounded vowels /i, e, EJ/, e.g. *geyla* ‘yell’ [‘dZEJ:la]. The letter combinations *gj, dj, kj, tj* and *hj* also have alveopalatal sounds, e.g. *tjóvur* ‘thief’ [‘tSOW:wUr], and when /s/ is followed *by j, kj, tj or k* followed by the aforementioned front unrounded vowels, /s/ is palatalized as well, e.g. *skip* ‘ship’ [‘Si:b]. These are all general rules, but exceptions to the rule exist.

Table 4 shows some combinations of vowels and consonants that have an unexpected pronunciation. Note that in the list, we include the

[5] This is not true for the dialect of Suðuroy.

[6] /r/ can have multiple pronunciations in Faroese, including alveolar and trill. We have opted for the most frequent pronunciation, namely the alveolar.

	Long vowel		Short vowel	
Grapheme	**Phoneme**	**Example**	**Phoneme**	**Example**
á	[OA:]	*gráur* ‘grey’	[O]	*grátt* ‘grey (n.)’
a	[EA:], [a:] (in loanwords)	*glaður* ‘happy’, tomat ‘tomato’	[a]	*glatt* ‘happy (n.)’
æ	[EA:]	*læra* ‘teach’	[a]	*lærdi* ‘taught’
e	[e:]	*meta* ‘estimate’	[E]	*metti* ‘estimated’
i	[i:]	*fita* ‘fatten’	[I]	*fitna* ‘get fat’
y	[i:], [y:]	*fyri* ‘for, before’, *myta* ‘myth’	[I], [Y]	*fyrr* ‘earlier’, mystiskur ‘mythical’
í	[UJ:]	*lítil* ‘small’	[UJ]	*lítli* ‘small (def. m.)
ý	[UJ:]	*sýta* ‘refuse’	[UJ]	*sýtti* ‘refused’
o	[o:]	*tosa* ‘talk’	[O]	*toldi* ‘endured’
ó	[OW:]	*rópa* ‘yell’	[9]	*rópti* ‘yelled’
u	[u:]	*gulur* ‘yellow’	[U]	*gult* ‘yellow (n.)’
ú	[3W:]	*púra* ‘quite, entirely’	[Y]	*púrt* ‘quite, entirely’
ø	[2:]	*søtur* ‘sweet’	[9]	*søtt* ‘sweet (n.)’
ei	[aJ:]	*heitur* ‘warm’	[aJ]	*heitt* ‘warm (n.)’
ey	[EJ:]	*reyður* ‘red’	[E]	*reytt* ‘red (n.)’
oy	[OJ:]	*royna* ‘try’	[OJ]	*royndi* ‘tried’

Table **3**: The different graphemes representing vowels in Faroese and their phonemic counterparts. Note that length is not represented in our G2P-dictionaries.

pronunciations relevant to the central dialect only (see Section 5 for more on dialects).

3.2 Some phonological considerations

Most descriptions of Faroese state that the phonemes /lmnr/ are devoiced before /ptk/ and in front of /s/ as well (e.g. Thráinsson et al., 2012). However, recent work in progress sheds doubt on this traditional description of devoicing in front of /s/. Lamhauge (2022) finds a dialectal difference between the northern and southern dialects of Faroese, suggesting that these sonorants are not as categorically devoiced as the literature describes them. However, the Ravnur project chose to phonetically transcribe the sonorants in front of /s/ as voiced sonorants.

Furthermore, when /s/ is followed by a short diphthong ending in a high vowel, i.e. the graphemes *í, ý, ei*, and *oy*, the /s/ can be pronounced as [S] (Lockwood, 2002), e.g. píska ‘whip’. However, in this condition, /s/ is transcribed [s] in the Ravnur lexicon.

Grapheme combination	**Phoneme**	**Example**
-ógv-	[Egv]	*krógv* ‘inn’
-úgv-	[Igv]	*búgv* ‘home’
-ang-	[ENg]	*svangur* ‘hungry’
-angi-	[EndZI]	*svangir* ‘hungry
-ank-	[ENg]	*blanka* ‘polish’
-eingi- *-einki-*	[OJndZI] [OJxdZI]	*dreingir* ‘boys’, *einki* ‘nothing’
hv-	[kv]	*hvør* ‘who’
-ll(-)[7]	[dl]	*øll* ‘everyone’
-rn[8]	[dn]	*bjørn* ‘bear’
-nn- after *ei, oy*	[dn]	*seinni* ‘later’
-ðr-	[gr]	*veðrur* ‘ram’
-ðg-	[g:]	*steðga* ‘stop’
-ðk-	[hk:]	*blíðka* ‘make gentle’
-gd-	[d:], [gd]	*løgdu* ‘laid (pl.), *løgd* ‘laid (f.)’
-vd-	[d:], [Wd]	*høvd* ‘head’, *høvd* ‘rise’, resp.
-rs-	[z]	*mars* ‘March’
-um (unstressed)	[Un]	monnum ‘men’

Table **4**: Grapheme combinations and their phonemic representations.

[7] In loanwords, *-ll(-)* is pronounced as [l:], e.g. *ball* ‘party’.

[8] When /r+n/ are combined in the inflection of a word, it is pronounced [rn], e.g. *far+nir* (m.pl. of *farin*). There are other exceptions to the /r+n/ rule as well (Thráinsson et al. 2012).

The Ravnur project decided to make the pronunciations in the dictionary distinct, phonological pronunciations. This means that the unstressed vowels [I] and [U] are transcribed as such, even though they have merged to [8] in central Faroese spontaneous speech (Petersen, 2022). A word like *mammu* 'mother (acc.)' would therefore be pronounced [mam:8] in the Central dialect, but is transcribed as [mam:U] in the Ravnur dictionary. Also, the unstressed syllables –arnir, -arnar, -irnar and –urnar are fully phonetically transcribed, e.g. *hundarnir* 'the dogs' ['hUndarnIr], although these are frequently pronounced without an /r/ in spontaneous speech, e.g. [hUndanIr] (Adams and Petersen, 2014).

These decisions made by the original Ravnur group have been followed in both our CENTRAL and EAST dictionary. Research remains to show how these phonological conditions are actually produced, and whether or not there is dialectal difference.

4 Representing pronunciation variations in one form

For developing the G2P-tool, we used the open source pronunciation dictionary from project Ravnur (Debess et al., 2019; Simonsen et al., 2022). However, we only use one single pronunciation per word. The original dictionary had many entries with multiple pronunciations, and for this project of developing the G2P-tool we had to prioritize just one of them. Being a descriptive dictionary, some words were assigned multiple pronunciations due to language variation (differences in dialect, sociolect or other lects) and assimilation of different kinds. In cases of dialectal variations, the pronunciations of the central dialect were chosen, as the different dialects are represented through separate dictionaries. In cases of other variation with no research to base the decision on, the choice was based on (by the Faroese linguist) perceived frequency of the forms, choosing the more frequent as the primary - knowing the limitations of this method.

Quite many of the word forms with multiple pronunciations were due to homographs. As the dictionary version for the G2P tool only operates with the values of orthography and pronunciation and no grammatical or semantic information, homographic word forms that belong to different lexemes melt into one, with their respective pronunciations being registered as pronunciation variations of the same form. As this tool strictly focuses on the relationship between graphs and phones, the semantics and grammar to differentiate homographic word forms are of course obsolete, but the omission of these values also presents challenges. Choosing one pronunciation over another in these cases, where both forms are considered to be valid in Faroese language, blurs the depiction of the linguistic reality, but is necessary for this linear conversion tool and might even increase accuracy of the tool.

In cases of multiple pronunciations due to homographs, one pronunciation has been chosen to be primary based on the main criteria of frequency and second phonological heritage (detailed below), having functionality and error rates of the tool in mind.

Frequency derived from searches in available corpora or web-search-based frequency was not always sufficient due to the relatively small volume of the resources, many of them not tagged, making it difficult to distinguish homographic forms. In these cases, other frequency measures were also taken into account:

- Native speaker intuition. Example: *havi* 'to have' PRES.1.SG /hEAvI/ > *havi* 'garden' SG.NOM /ha:vI/.
- Function words > content words.
- Grammatical case of the word form (the genitive is very rare (Thráinsson et al., 2012), and the pronunciation pairs were often due to a genitive word form opposite a word form with another grammatical case or from another part of speech). Example: *loksins* 'finally' (adverb) /lOgsIns/ > *loksins* 'lid' SG.GEN.DEF /lo:gsIns/.
- Frequent > obscure inflectional forms.

Though frequency being the main criteria, we also took into account and implemented rules of phonological heritage:

- Words of linguistic heritage > loan words (even though the loan word is well implemented in Faroese). This ensures the systematic phonological rules of Faroese a broader representation in the data. Example: *banki* 'to knock' PRES.1.SG /bExdZI/ > *banki* 'bank' SG.NOM /baxdZI/
- Common nouns > proper nouns. Proper nouns in general follow the phonological

rules to a lesser extent than common nouns. Examples: *allan* 'all' SG.MASC.ACC.DEF [adlan] > *Allan* 'Alan' (person name) [alan].

5 Two dictionaries for Faroese

The original dictionary from the Ravnur project is based on the dialect of the capital in the Faroe Islands. Even though there is no official standard dialect for Faroese (Petersen, 2022), there is believed to be some kind of central Faroese based partly on the dialect of the capital and partly on the written language (Jacobsen, 2011; Knooihuizen, 2014). However, in working with this G2P tool, we wanted to have greater diversity and decided to make two versions of the original dictionary reflecting two different dialect areas. The two dialect areas are the central dialect area, where the capital is located, called CENTRAL henceforth, and part of the northwest dialect area, called EAST henceforth (see Figure 1). The CENTRAL dialect area has the largest number of inhabitants, and the EAST dialect area has the second highest number of inhabitants. Combined, these two dialect areas comprise around 71% of the population[9].

The islands in the northwest area are for several reasons classified as being the same dialect area in the most recent dialect classification (Petersen, 2022). However, there is one important phonetic difference between the westernmost islands and the more central and eastern islands in that dialect area, and therefore, the westernmost part of the dialect area is not included in our EAST dictionary (the dotted line in Figure 1 marks the two parts of the northwest dialect area). This phonetic difference is the pronunciation of the digraph *ei*. For the same reason, we have given this dictionary the name EAST. This way, it is possible to make WEST, NORTHERN and SOUTHERN dictionaries as well, should the possibility present itself.

In Section 3, we presented the general phonological rules for (central) Faroese. In the following section, we will outline the main dialectal differences between the CENTRAL and the EAST dictionaries. For further information on dialectal differences in Faroese, see Thráinson et al. (2012) and Petersen (2022).

5.1 Phonological differences between CENTRAL and EAST

The main dialectal differences between the CENTRAL and EAST dialect area are as follows:

1) The letter *ó* is pronounced [OW] in the central dialect area and as [9W] in the east dialect area, 2) the digraph *ei* is pronounced [aJ] in the central area and as [OJ] in the east area, and 3) lack of preaspiration after long, non-high vowels before fortis[10] stop closures in the central area and preaspirated stop closures in the same condition in the east area (Petersen, 2022).[11] We will go through each of these three features in turn and explain how we did the changes, exceptions to the rule etc.

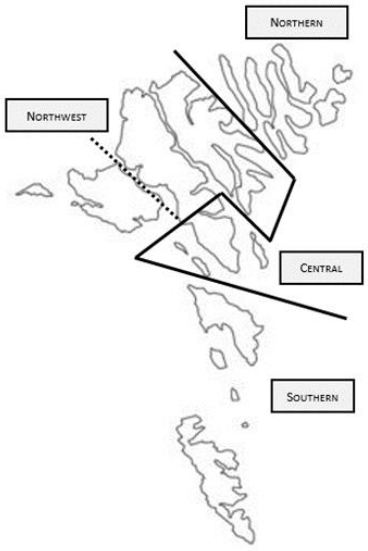


Figure **1**: The four main dialect areas based on Petersen (2022). The area to the west of the dotted line is the area from the northwest dialect area that is not part of our EAST dictionary.

Variation in pronunciation of *ó*

The CENTRAL dictionary being the starting point for creating an EAST dictionary, we needed to convert all [OW] sequences in the dictionary to [9W] sequences. The conversion was straight forward. Exemptions to this conversion were 1)

[9] https://hagstova.fo/fo/folk/folkatal/folkatal

[10] The terms 'fortis' and 'lenis' are often used on an abstract level to distinguish between the stop series /ptk/ and /bdg/ (Helgason 2002; Hejná 2015; among others). In using these terms, we are simply following this tradition, not implying any specific phonetic differences between the two series. Others call these series hard and soft, respectively (Thráinsson et al. 2012).

[11] As the phonetics of Faroese in general are understudied (Lamhauge 2022), there might be other dialectal differences that have not been described or studied yet, as is shown in Lamhauge (2022) in the case of sonorant devoicing. In cases of doubt, we have followed the decisions made by the original Ravnur project.

loan words and 2) words with the orthographic sequence -ov-. Loan words or foreign words (especially from English), which have not been implemented enough in Faroese language to adapt to dialectal differences in pronunciation, were not converted, e.g. Windows [vIndOWs] (proper noun) and karaoke [karaOWki].

The letter sequence *-ov-* can also manifest in an [OW]-pronunciation in Faroese, e.g. in *bakarovnur* 'oven' SG.NOM [bEAgarOWnUr] and *flovislig* 'embarrassing' SG.FEM.NOM [flOWwIsli]. The -ov- based [OW] has no dialectal variation and is pronounced the same throughout all the dialects. In the conversion process of [OW] to [9W], words with the orthographic sequence *-ov-* together with an [OW] in pronunciation were therefore exempt, and their transcriptions were not converted.

The pronunciation of phonologically short *ó* is not relevant for distinguishing CENTRAL and EAST and is not discussed further here.

Variation in pronunciation of *ei*

The pronunciation of the phonologically long digraph *ei* is dialectally distributed as mentioned in the previous section. We converted all [aJ] sequences in CENTRAL to [OJ] sequences in EAST. Exemptions to this conversion were 1) loan words and 2) proper nouns. Loan words or foreign words (especially from English), which have not been implemented enough in Faroese language to adapt to dialectal differences in pronunciation were not converted. This applies to loan words, e.g. *gurkumeia* 'turmeric' SG.NOM [gUzgUmaJja] or *bei* 'bye' [baJ][12]. Proper nouns with long *ei* behave quite differently regarding pronunciation than other parts of speech, and even though there is a variation in the pronunciation (between [aJ] and [OJ]), the variation seems to be non-systematic and idiolectal. For this reason, proper nouns were exempt from the conversion process. The exemptions of loan words and proper nouns were done manually.

Variation in preaspiration

Table 5 shows an overview of the phonological conditions in which preaspiration occurs in different Faroese dialects. In some phonological conditions, preaspiration occurs in all the dialects, i.e. between a short vowel and a long stop closure, or after a short vowel and before a stop closure followed by a sonorant. However, between a long, non-high vowel and a short consonant, only some dialects preaspirate. If the vowel is high in this condition, none of the dialects preaspirate (Thráinsson et al., 2012). There are other phonological conditions as well, in which preaspiration might or might not occur. However, as these have not been studied sufficiently, they are not included in this overview.[13]

Phonological condition	**Example**	**Dialects**
$VC{:}_{(ptk)}$	*Átta 'eight'* [OHd:a]	All dialects
$VC_{(ptk)}C_{(mn)}$ $VC_{(t)}C_{(l)}$	*Vatn* 'water' [vaHdn]	All dialects
$V{:}_{(non\text{-}high)}C$	*Kaka* 'cake' [kEA:Hga]	Only some dialects (including EAST, excluding CENTRAL)

Table **5**: An overview of the phonological conditions in which preaspiration occurs in different Faroese dialects (based on Thráinsson et al., 2012).

This means that in the CENTRAL dictionary, there is no preaspiration in the V:C condition. We therefore had to insert preaspiration in this condition in the EAST dictionary. As the CENTRAL dictionary does not have length marks neither on vowels nor on consonants, the process could not be done automatically. We searched for all of the relevant phonological conditions, but as there are quite a few exceptions to the rules, i.e. in loanwords, all of the instances had to be checked manually before implementing the change. Furthermore, as there is actually not much known about the pronunciation of stops in loanwords, we have made educated guesses based on native speaker intuitions at times. For example,

[12] Words with the [aJ] pronunciation can also be found with other spelling than *ei*, e.g. *wifi* 'wifi' [vaJfaJ] and *kai* 'dock' SG.NOM [kaJ], but this is due to the foreign heritage of the words. All words with [aJ] and other spelling than *ei* were also exempt.

[13] For example, Lamhauge (2022) has found in her work in progress a dialectal split in the pronunciation of non-homorganic stops. In these cases, we have followed decisions made by the original Ravnur project, and we have made no changes to these decisions in the EAST dictionary. See Thráinsson et al. (2012) and Helgason (2002) for a more thorough overview of preaspiration in Faroese.

in loanwords that end in –at, we have chosen to insert preaspiration, e.g. salat ‘salad’ [‘salaHd]. In other cases, the decision was made on a word basis.

6 G2P

The "Faroese G2P Models" is a set of two models trained with the software tool "Sequitur-G2P" (for details on Sequitur-G2P, see Bisani and Ney, 2008), which is a trainable grapheme-to-phoneme tool developed at RWTH Aachen University (https://www.rwth-aachen.de/) by Maximilian Bisani. One of the models is for Central Faroese and the other is for the East variant.

The set also includes two files with the corresponding repertory of phonemes for the Central (central.phones) and the East (east.phones) variants.

6.1 The training set

In order to train the Sequitur-G2P, it is necessary to provide it with a pronunciation dictionary that will play the role of training set. This pronunciation dictionary was taken from BLARK 1.0 and double-checked by experts and finally split into one set for the Central and East variants of Faroese. The characteristics of both training dictionaries are the following:

- Both dictionaries contain 197,757 unique words each.
- Words with symbols other than letters of the Faroese alphabet (e.g. *kt-vinnu*, *stóra_dímun*) were excluded from the training dictionaries because those symbols do not have a correspondence in phonemes.
- The pronunciations are based on the FARSAMPA alphabet provided in the BLARK 1.0.
- Multiple pronunciations for one particular word are not accepted. Therefore, there is only one pronunciation associated with each word. This is to avoid providing Sequitur with inconsistencies (see more in Section 5)[14].
- The number of phonemes for the Central variant is 60, which is a subset of the East variant, which has 63.

Most of the entries in both dictionaries are the same, the only difference occurs with words that can be pronounced differently in both variants.

As the Sequitur models are destined to do ASR experiments, the diacritics for length (:), primary stress (%), secondary stress (~) and emphasis (!) were not taken into account, because we saw that they do not offer any advantage to the ASR experiments but they make the models unnecessarily more complex instead[15].

6.2 Evaluation and results

In order to evaluate the performance of the Sequitur models, a set of 1000 words with pronunciation was randomly selected for each variant. The resulting test sets do not contain the same words, and the test sets are not included in the training dictionaries.

The evaluation was performed using the evaluation command provided by Sequitur. Table 1 shows a summary of the results obtained:

	Central Model	**East model**
Total	1000 strings, 9703 Symbols	1000 strings, 9615 symbols
Successfully translated	100% strings, 100% symbols	100% strings, 100% symbols
string errors	22 (2.20%)	31 (3.10%)
symbol errors	29 (0.30%)	44 (0.46%)
insertions	7 (0.07%)	4 (0.04%)
deletions	3 (0.03%)	5 (0.05%)
substitutions	19 (0.20%)	35 (0.36%)
translation failed	0% strings, 0% symbols	0% strings, 0% symbols
total string errors	22 (2.20%)	31 (3.10%)
total symbol errors	29 (0.30%)	44 (0.46%)

Table **8**: Evaluation results obtained from Sequitur's evaluation command

As can be seen in Table 8, the translation errors are below 5% in both models, indicating that the models are reliable and not far away from other models found in the literature (Milde et al., 2017).

[14] This is also explains why we trained one model for each dialect, instead of training one joint model and apply rules as a post-processing step.

[15] In the case of TTS, it would be beneficial to train a different model for that purpose that includes the diacritics.

7 Conclusion

A standardized pronunciation dictionary of good quality is crucial for the development of spoken language technologies. This work describes the definition, development and the establishment of a Faroese pronunciation dictionary and a grapheme-to-phoneme tool to go along with it. This work is also important in understanding spoken Faroese and can be used to study regional differences in accents and dialects. It is clear that this will form a basis of further studies of Faroese and development of spoken language technologies such as ASR and TTS.

References

Anna B. Nikulásdóttir, Jón Guðnason, and Eiríkur Rögnvaldsson. 2018. An Icelandic pronunciation dictionary for TTS. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 339-345.

Annika Simonsen, Sandra S. Lamhauge, Iben N. Debess, and Peter J. Henrichsen. 2022. Creating a basic language resource kit for faroese. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4637-4643.

Anton K. Ingason, Eiríkur Rögnvaldsson, Einar F. Sigurðsson, and Joel C. Wallenberg. 2012. Faroese Parsed Historical Corpus (FarPaHC) 0.1, CLARIN-IS, http://hdl.handle.net/20.500.12537/92.

Benjamin Milde, Christoph Schmidt, and Joachim Köhler. 2017. Multitask Sequence-to-Sequence Models for Grapheme-to-Phoneme Conversion. In *INTERSPEECH*, pages 2536-2540.

Bente Maegaard, Steven Krauwer, Khalid Choukri, and Lise D. Jørgensen. 2006. The BLARK concept and BLARK for Arabic. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 773- 778.

Carlos D. Hernández Mena. 2022a. Kaldi Recipe for Faroese, CLARIN-IS, http://hdl.handle.net/20.500.12537/305.

Carlos D. Hernández Mena and Annika Simonsen. 2022b. Ravnursson Faroese Speech and Transcripts, CLARIN-IS, http://hdl.handle.net/20.500.12537/276.

Carlos D. Hernández Mena, Sandra S. Lamhauge, Iben N. Debess, and Annika Simonsen. 2022c. Faroese Language Models with Pronunciations, CLARIN-IS, http://hdl.handle.net/20.500.12537/304.

Hjalmar P. Petersen. 2021. *Føroysk mállæra 3: Ljóðlæra*. Nám, Tórshavn, Faroe Islands.

Hjalmar P. Petersen. 2022. Evidence for the modification of dialect classification of modern spoken Faroese. *European Journal of Scandinavian Studies*, 52(1):43-58.

Höskuldur Thráinsson, Hjalmar P. Petersen, Jógvan í L. Jacobsen, and Zakaris S. Hansen. 2012. *Faroese, an overview and reference grammar*, 2. ed. Fróðskapur, Tórshavn, Faroe Islands, and Linguistic Institute, University of Iceland, Reykjavík, Iceland.

Iben N. Debess, Sandra S. Lamhauge, Annika Simonsen, Peter J. Henrichsen, Egil Hofgaard, Uni Johannesen, Petur Markus J. Hammer, Gunnvør H. Brimnes, Ebba Malena D. Thomsen, and Beinta Poulsen. 2022. Basic Language Resource Kit 1.0 for Faroese. OpenSLR.org, https://www.openslr.org/125/

Iben N. Debess, Sandra S. Lamhauge, and Peter J. Henrichsen. 2019. Garnishing a phonetic dictionary for ASR intake. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics, NODALIDA 2019*, pages 395-399.

Jonathan Adams and Hjalmar P. Petersen. 2014. *Faroese: A Language Course for Beginners*, 3rd ed.). Stiðin, Tórshavn, Faroe Islands.

Jógvan í Lon Jacobsen. 2011. Dialektbrugere i spændetrøje? – om dialektbrug i et lille, tæt sprogsamfund. In Gunnstein Akselberg & Edit Bugge (Eds.), *Vestnordisk språkkontakt gjennom 1200 år*, pages 181-200. Fróðskapur.

Maximilian Bisani, and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech communication*, 50(5):434-451. GitHub: https://github.com/sequitur-g2p/sequitur-g2p

Michaela Hejná. 2015. Pre-aspiration in Welsh English: A case study of Aberystwyth. PhD thesis, University of Manchester, Manchester.

Pétur Helgason. 2002. Preaspiration in the Nordic languages: synchronic and diachronic aspects. PhD thesis, Stockholm University, Stockholm.

Remco Knooihuizen. 2014. Variation in Faroese and the development of a spoken standard: In search of corpus evidence. *Nordic journal of linguistics, 37*(1):87-105.

Robert Weide. 1998. The Carnegie Mellon pronouncing dictionary. release 0.6, www.cs.cmu.edu.

Sandra S. Lamhauge. 2022. Preaspiration and sonorant devoicing in two Faroese dialects. Talk presented at Phonetics and Phonology in Denmark 2022, 17-18 November 2022, University of Copenhagen.

Steven Krauwer. 2003. The Basic Language Resource Kit (BLARK) as the First Milestone for the Language Resources Roadmap. In *Proceedings of the International Workshop "Speech and Computer", SPECOM 2003*, pages 8-15, Moscow, Russia.

Tony Robinson. BEEP Dictionary. OpenSLR, https://www.openslr.org/14/

William B. Lockwood. 2002. *An Introduction to Modern Faroese*, 4th ed. Føroya skúlabókagrunnur, Tórshavn, Faroe Islands

Using Membership Inference Attacks to Evaluate Privacy-Preserving Language Modeling Fails for Pseudonymizing Data

Thomas Vakili and Hercules Dalianis
Department of Computer and Systems Sciences (DSV)
Stockholm University, Kista, Sweden
{thomas.vakili,hercules}@dsv.su.se

Abstract

Large pre-trained language models dominate the current state-of-the-art for many natural language processing applications, including the field of clinical NLP. Several studies have found that these can be susceptible to privacy attacks that are unacceptable in the clinical domain, where personally identifiable information (PII) must not be exposed.

However, there is no consensus regarding how to quantify the privacy risks of different models. One prominent suggestion is to quantify these risks using membership inference attacks. In this study, we show that a state-of-the-art membership inference attack on a clinical BERT model fails to detect the privacy benefits of pseudonymizing data. This suggests that such attacks may be inadequate for evaluating token-level privacy preservation of PIIs.

1 Introduction

State-of-the-art results in natural language processing typically rely on large pre-trained language models (PLMs) such as BERT (Devlin et al., 2019) or models in the GPT family (Radford et al., 2019). Multiple studies have found that their large number of parameters can cause PLMs to unintentionally memorize information in their training data, making them vulnerable to privacy attacks (Carlini et al., 2019, 2021). At the same time, other studies have shown that training PLMs using domain-specific data yields better results on domain-specific tasks (Lee et al., 2020; Lamproudis et al., 2021). In the clinical domain, these combined findings pose a significant challenge: training PLMs with clinical data is necessary to achieve state-of-the-art results. However, PLMs can be vulnerable to privacy attacks that are especially dangerous when training with clinical data. Broadly speaking, these attacks can be divided into two classes: training data extraction attacks and membership inference attacks.

1.1 Privacy Attacks

Training data extraction attacks are the more severe class of attacks. An adversary who successfully mounts such an attack can extract details about training data that were used to train a PLM. Carlini et al. (2021) show that GPT-2 is vulnerable to such attacks. Several studies (Nakamura et al., 2020; Lehman et al., 2021; Vakili and Dalianis, 2021) have tried to mount similar attacks on BERT models. To this date, there are no examples of successful training data extraction attacks targeting BERT models.

Membership inference attacks (MIAs) do not aim to *extract* training data from models. Instead, these attacks try to discern whether or not a datapoint was present in a model's training data. Inferring that a datapoint has been present in the training data is less severe than extracting it but could, for example, reveal if a patient has visited a set of clinics.

MIAs have been proposed as a proxy for measuring the degree of memorization in machine learning models (Shokri et al., 2017; Murakonda and Shokri, 2020; Mireshghallah et al., 2022). Both training data extraction attacks and MIAs rely on some degree of memorization in the model. However, MIAs do not require any algorithms that generate the memorized data. By focusing solely on detecting memorization, MIAs are used to estimate a worst-case degree of privacy leakage. Indeed, MIAs are the basis for the ML Privacy Meter developed by Murakonda and Shokri (2020).

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 318–323

1.2 Protecting Datapoints or Tokens?

One special property of natural language data is that many words in a sentence can be replaced with synonyms without changing the overall semantics of the sentence. This feature is interesting from a privacy perspective and is the basis for *pseudonymization*.

Pseudonymization is the process of replacing sensitive information with realistic surrogate values. For example, names are replaced with other names or with placeholders. These kinds of sensitive words or phrases are rarely important for the utility of the data, neither for fine-tuning models (Berg et al., 2021; Vakili and Dalianis, 2022), pre-training models (Verkijk and Vossen, 2022; Vakili et al., 2022), nor for general research purposes (Meystre et al., 2014a,b). One important example of this is MIMIC-III (Johnson et al., 2016), which contains a large number of electronic health records in which sensitive words or phrases have been manually replaced with placeholders. This dataset is widely employed in clinical machine learning and is considered to be relatively safe.

One fundamental assumption of pseudonymization is that the higher-level semantics of a text are not important from a privacy perspective. For example, an electronic health record describing a patient visiting a hospital is not sensitive if we cannot infer *who* the patient is, *when* the visit took place, and so on. One way of viewing this is that the data are not primarily sensitive on the datapoint level, but on the token level.

1.3 Membership Inference Attacks and Pseudonymization

Manual pseudonymization is a time-consuming process. Many institutions lack the resources to manually pseudonymize data on the scale required for modern machine learning models or even for less data-intensive qualitative clinical research. An alternative is to use *automatic pseudonymization*. Automatic pseudonymizers typically rely on named entity recognition (NER) to detect sensitive information. The detected entities are then either replaced with realistic surrogates or with placeholders. However, NER systems are rarely perfectly accurate. Imperfect recall leads to some sensitive entities remaining after processing the data, which is undesirable from a privacy perspective.

Because systems performing automatic pseudonymization fail to detect some sensitive entities, it is important to measure the privacy implications of this. A straightforward approach is to consider the recall of the NER model that powers the system. This metric can be used to estimate the number of sensitive entities that remain in the data. Such estimates are useful for determining the sensitivity of an automatically pseudonymized dataset. However, they are less ideal for judging the privacy risks of a machine learning model trained using the dataset. Assuming that the trained model has memorized every single sensitive entity is overly pessimistic.

Estimating the privacy risks of models using MIA, as suggested by Mireshghallah et al. (2022), is an attractive alternative that would allow pseudonymization to be compared to other privacy-preserving techniques. However, MIAs are designed to measure the memorization of entire datapoints rather than the memorization of sensitive tokens. This poses a challenge to the paradigm of using MIAs to estimate the privacy risks of machine learning models trained using pseudonymized data.

In this study, we show that the state-of-the-art MIA described by Mireshghallah et al. (2022) cannot distinguish between a model trained using real or pseudonymized data. These results suggest that using this attack to quantify privacy risks fails to capture privacy gains from pseudonymizing training data.

2 Methods and Data

This study closely mirrors the experimental setup used by Mireshghallah et al. (2022) in order to minimize discrepancies stemming from differences in implementation details. The datasets and models are based on resources introduced by Lehman et al. (2021). The experiments aim to examine whether or not membership inference attacks can distinguish between a model trained using real or pseudonymized data.

2.1 Data

This study uses the ClinicalBERT-1a model trained by Lehman et al. (2021). They train a model using pseudonymized clinical notes from a subset of MIMIC-III. This specific model is of the same size as BERT-base (Devlin et al., 2019) and uses this model's parameters as a starting point for continued pre-training to adapt the model to the clinical domain. The corpus used to train

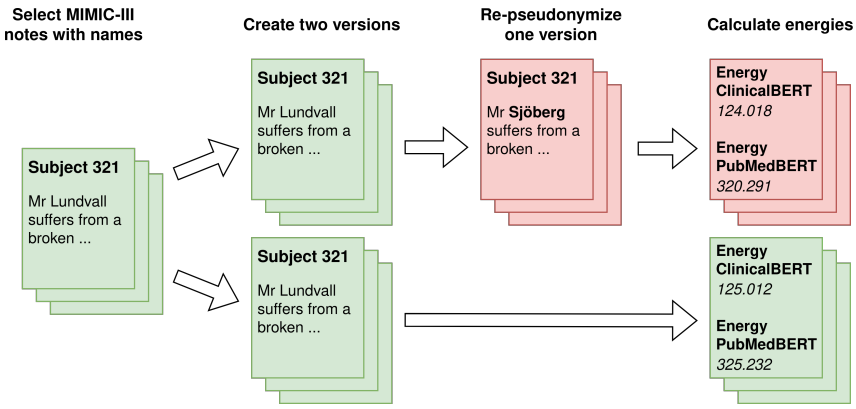


Figure 1: Our experiments use a filtered subset of MIMIC-III that only contains records with named (but pseudonymized) patients. One subset, the *Pseudo* subset, has been used to create the ClinicalBERT model used as the target for the attack. Another version, referred to as the *Real* dataset, is re-pseudonymized and acts as a stand-in for the original sensitive raw data.

the model is also available. Mireshghallah et al. (2022) perform their membership inference experiments using the training data for the BERT model and MIMIC-III data that was not used for training the model. The method also needs a reference model, and this study follows their example by also using PubMed-BERT (Gu et al., 2021) for this purpose.

This study focuses specifically on MIAs' ability to discern whether or not a model has been trained using pseudonymized data. A filtered version of MIMIC-III containing only sentences with names is created to ensure that the results reflect this distinction. This dataset contains a total of 236,114 datapoints. A pseudonymized version of the dataset is created in which all names have been replaced with other names.

After replacing all the names, we have two datasets where each sentence differs solely in what names are used. The dataset used to train the model will be referred to as the *Pseudo* dataset, and the re-pseudonymized dataset will be referred to as the *Real* dataset. This mimics the situation where we have a model trained on *perfectly pseudonymized* training data. Figure 2 illustrates the scenario that is simulated. Ideally, the membership inference attack should indicate that replacing all names with pseudonyms has made the model much safer.

2.2 Predicting Membership

This study uses the same procedures as Mireshghallah et al. (2022) since their method is the current state-of-the-art membership inference attack targeting masked language models like BERT. The method works by analyzing how the target model reacts to a datapoint as compared to a reference model. The target and reference models, in our case ClinicalBERT and PubMed-BERT, differ in that the target model has been trained using sensitive data that the reference model has not been exposed to.

A variety of different measurements can represent the reaction of the model. Following the example of Mireshghallah et al. (2022), we use the *normalized energy values* calculated for every datapoint. These values $E_\theta(S)$ are calculated by estimating the probability of a sequence of tokens S given a set of masking patterns M for a model with the parameters θ:

$$E_\theta(S) = \frac{1}{|M|} \sum_{m \in M} e_\theta(S, m)$$

$$e_\theta(S, m) = \sum_{i \in m} \log\Big[p_\theta(S_i \mid S_m)\Big]$$

S_i is the token at index i and S_m is the altered sequence S to which the masking pattern m has been applied. These normalized energy values are calculated for three datasets, for both the target model and the reference model:

In-data Parts of the dataset used to train the target model. In this study, the two datasets described in Section 2.1 fill this function, as illustrated in Figure 1.

Out-data A second dataset known *not* to belong to the target models training data. This subset

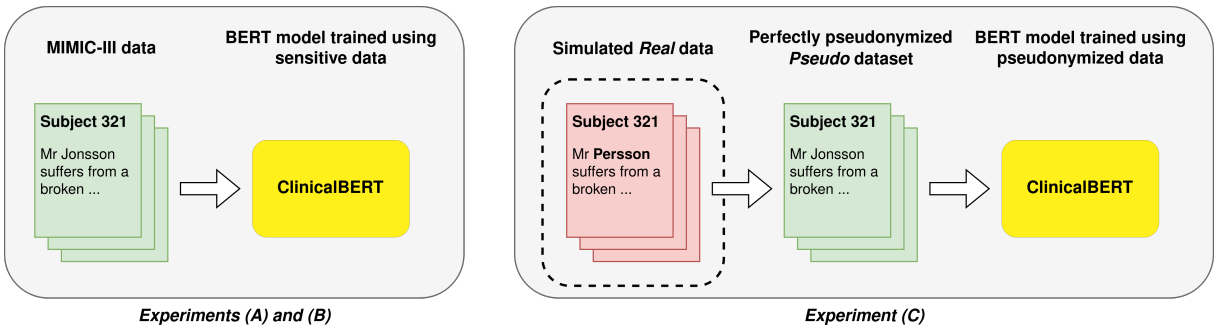


Figure 2: This study simulates the scenario in which a perfectly pseudonymized dataset has been used for continued pre-training of a BERT model. The version of MIMIC-III used to create the ClinicalBERT model from Lehman et al. (2021) is re-identified with pseudonyms and is used in experiments (A) and (B). We refer to this dataset as the *Pseudo* dataset. In experiment (C), we simulate the original, pre-pseudonymized MIMIC-III by populating the data with other names and call this version the *Real* dataset.

of MIMIC-III is also used in Mireshghallah et al. (2022).

Threshold data A third dataset disjoint from the *Out-data* and known not to belong to the target models training data. A subset of i2b2 (Stubbs and Uzuner, 2015) is used, as in Mireshghallah et al. (2022).

The normalized energy values of the target and reference models are compared for the threshold data, resulting in a threshold. This threshold is used to classify if a datapoints belongs to the *In-data* or the *Out-data* based on the difference between the energy values of the datapoint obtained from the models. The intuition behind this method is that if the target model has memorized a datapoint, then its energy value will be noticeably higher relative to the reference model's energy value. The threshold is set so that 90% of the datapoints in the *threshold dataset* are correctly classified as non-members (Mireshghallah et al., 2022). We also calculate the AUC to provide a threshold-independent assessment of the privacy risks.

This study examines the claim that membership inference attacks can be used to quantify privacy gains from using various privacy-preserving techniques. The scenario modeled in these experiments simulates the situation where the privacy-preserving technique is perfect pseudonymization. Every datapoint with a named patient in the training data for ClinicalBERT has a corresponding datapoint in the *Real* dataset where the name is different. In such a scenario, no real names are left in the training data to memorize. Thus, the risk of leaking any name of a patient is zero, representing a substantial increase in privacy. If the attack accurately quantifies these privacy gains, then we would expect it to perform worse when the data has been pseudonymized.

3 Results

Three different attacks are performed using three different datasets as the in-data. The accuracy, precision, and recall values of each attack are listed in Table 1. Experiment (A) mirrors the setup used by Mireshghallah et al. (2022). Experiments (B) and (C) use the subsets of MIMIC-III that only contain names. There are only very small differences in the correctness of the classifications, regardless of the configuration used.

Table 1 also lists the AUC, which represents a threshold-independent evaluation of the MIAs. The AUC varies more than the other three metrics. However, the difference between experiments (A) and (B) is larger than that between experiments (B) and (C). This is despite the fact that the *In-data* for experiments (A) and (B) come from the same population. The difference in AUC between experiments (B) and (C) is 0.017.

Experiments (A) and (B) represent cases where we have not performed any pseudonymization of the training data. That is, the *In-data* are used to train the BERT model without employing any privacy-preserving techniques. Experiment (C) is the result of the simulated scenario where perfect pseudonymization is employed to preserve the privacy of the data. In other words, the model is not exposed to any real names during training. The privacy gains from using this technique are not reflected by the metrics in Table 1.

	In-data	Out-data	Threshold	Accuracy	Precision	Recall	AUC
(A)	*Pseudo, random sample*	*Held-out*	*i2b2*	*0.771*	*0.990*	*0.548*	*0.916*
(B)	Pseudo, names only	Held-out	i2b2	0.780	0.990	0.566	0.882
(C)	Real, names only	Held-out	i2b2	0.770	0.990	0.548	0.865

Table 1: The membership inference attack is run with three different configurations. Experiment (A) uses a random sample of MIMIC-III used in Mireshghallah et al. (2022) as in-data, and all experiments use the same out-data as they do. Experiments (B) and (C) use the datasets described in Section 2.1 for the in-data. The accuracy of each attack is displayed alongside the recall and the precision values. The threshold-independent AUC value is also listed.

4 Discussion and Conclusions

This study focuses specifically on protecting names. Future research would benefit from analyzing additional categories of PII. However, the data and models created by Lehman et al. (2021) focus specifically on names. This class of PII is used in this study to facilitate comparisons with earlier studies.

The results from the three experiments in Table 1 are very similar to each other. At the same time, experiment (C) represents a scenario in which a very strong privacy-preserving measure has been employed to increase the privacy of the target model. If the studied MIA is an accurate way of quantifying the privacy benefits of using pseudonymization, then we would expect the MIA to be much less accurate in experiment (C). The fact that the MIA works nearly as well for experiments (A) and (B) as for (C) indicates that using this attack to quantify memorization does so on a datapoint level. This may be useful for evaluating techniques such as differentially private pretraining (Li et al., 2022), which operate on entire datapoints.

It remains to be shown which of the datapoint's characteristics are used to separate members from non-members. The results of our experiments suggest that using this MIA does not accurately quantify the privacy gains from using pseudonymization, which instead operates on the token level. While the scope of this short paper was limited to evaluating a state-of-the-art MIA for BERT models, future research should also evaluate other MIAs and a wider range of privacy-preserving techniques.

Acknowledgements

We want to thank Fatemehsadat Mireshghallah for sharing the data and code used in Mireshghallah et al. (2022). We are also grateful to the DataLEASH project for funding the research presented in this paper.

References

Hanna Berg, Aron Henriksson, Uno Fors, and Hercules Dalianis. 2021. De-identification of Clinical Text for Secondary Use : Research Issues. In *Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2021) - Volume 5: HEALTHINF*, pages 592–599. SciTePress.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *Proceedings of the 28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting Training Data from Large Language Models. In *Proceedings of the 30th USENIX Security Symposium*, pages 2633–2650.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *ACM Transactions on Computing for Healthcare*, 3(1):2:1–2:23.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghas-

semi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):160035. Number: 1 Publisher: Nature Publishing Group.

Anastasios Lamproudis, Aron Henriksson, and Hercules Dalianis. 2021. Developing a Clinical Language Model for Swedish: Continued Pretraining of Generic BERT with In-Domain Data. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 790–797, Held Online. INCOMA Ltd.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron Wallace. 2021. Does BERT Pretrained on Clinical Notes Reveal Sensitive Data? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 946–959, Online. Association for Computational Linguistics.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2022. Large Language Models Can Be Strong Differentially Private Learners. In *Proceedings of the Tenth International Conference on Learning Representations*.

Stéphane Meystre, Shuying Shen, Deborah Hofmann, and Adi Gundlapalli. 2014a. Can physicians recognize their own patients in de-identified notes? *Studies in Health Technology and Informatics*, 205:778–782.

Stéphane M. Meystre, Óscar Ferrández, F. Jeffrey Friedlin, Brett R. South, Shuying Shen, and Matthew H. Samore. 2014b. Text de-identification for privacy protection: A study of its impact on clinical text information content. *Journal of Biomedical Informatics*, 50:142–150.

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. 2022. Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8332–8347.

Sasi Kumar Murakonda and Reza Shokri. 2020. ML Privacy Meter: Aiding Regulatory Compliance by Quantifying the Privacy Risks of Machine Learning. ArXiv:2007.09339 [cs].

Yuta Nakamura, Shouhei Hanaoka, Yukihiro Nomura, Naoto Hayashi, Osamu Abe, Shuntaro Yada, Shoko Wakamiya, and Eiji Aramaki. 2020. KART: Privacy Leakage Framework of Language Models Pre-trained with Clinical Records. *arXiv:2101.00036 [cs]*. ArXiv: 2101.00036.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. ISSN: 2375-1207.

Amber Stubbs and Özlem Uzuner. 2015. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus. *Journal of Biomedical Informatics*, 58:S20–S29.

Thomas Vakili and Hercules Dalianis. 2021. Are Clinical BERT Models Privacy Preserving? The Difficulty of Extracting Patient-Condition Associations. In *Proceedings of the AAAI 2021 Fall Symposium on Human Partnership with Medical AI: Design, Operationalization, and Ethics (AAAI-HUMAN 2021)*.

Thomas Vakili and Hercules Dalianis. 2022. Utility Preservation of Clinical Text After De-Identification. In *Proceedings of the 21st Workshop on Biomedical Language Processing at ACL 2022*, pages 383–388, Dublin, Ireland. Association for Computational Linguistics.

Thomas Vakili, Anastasios Lamproudis, Aron Henriksson, and Hercules Dalianis. 2022. Downstream Task Performance of BERT Models Pre-Trained Using Automatically De-Identified Clinical Data. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4245–4252, Marseille, France. European Language Resources Association.

Stella Verkijk and Piek Vossen. 2022. Efficiently and Thoroughly Anonymizing a Transformer Language Model for Dutch Electronic Health Records: a Two-Step Method. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1098–1103, Marseille, France. European Language Resources Association.

Sentiment Classification of Historical Danish and Norwegian Literary Texts

Ali Al-Laith[1,2], **Kirstine Nielsen Degn**[1], **Alexander Conroy**[1], **Bolette Sandford Pedersen**[1], **Jens Bjerring-Hansen**[1] **and Daniel Hershcovich**[2]
Department of Nordic Studies and Linguistics, University of Copenhagen[1]
Department of Computer Science, University of Copenhagen[2]
alal@di.ku.dk, kirstinenielsendegn@gmail.com, alc@hum.ku.dk,
bspedersen@hum.ku.dk, jbh@hum.ku.dk, dh@di.ku.dk

Abstract

Sentiment classification is valuable for literary analysis, as sentiment is crucial in literary narratives. It can, for example, be used to investigate a hypothesis in the literary analysis of 19th-century Scandinavian novels that the writing of female authors in this period was characterized by negative sentiment, as this paper shows. In order to enable a data-driven analysis of this hypothesis, we create a manually annotated dataset of sentence-level sentiment annotations for novels from this period and use it to train and evaluate various sentiment classification methods. We find that pretrained multilingual language models outperform models trained on modern Danish, as well as classifiers based on lexical resources. Finally, in the classifier-assisted corpus analysis, we both confirm and contest the literary hypothesis and further shed light on the temporal development of the trend. Our dataset and trained models will be useful for future analysis of historical Danish and Norwegian literary texts.

1 Introduction

Sentiment analysis, the computational study of emotions, opinions, and evaluations expressed in text, has become an important tool in natural language processing. It is based on the premise that words are associated with sentiments or valence, and that these associations can be quantified (Thomsen et al., 2021). However, its application in literary studies has been limited. Literary texts, such as novels, poems, and plays, provide a unique cultural window into past attitudes, beliefs, and emotions. By analyzing the sentiment expressed in these texts, researchers can gain a deeper understanding of the cultural and societal attitudes of the past and how they have shaped our present understanding of the world. However, traditional sentiment analysis techniques in the investigation of literary texts may be less effective due to the use of an archaic vocabulary, literary ambiguity, and figurative language, as well as the limited training data, the difficulties of generalizing models trained on modern text, and the challenges of annotation. Therefore, there is a need for new methods and approaches to analyze sentiment in historical and literary texts to uncover the valuable insights they can provide. However, there is no existing benchmark for sentiment analysis of the texts we are interested in, and it is not known how well existing, general methods perform on them, and whether they can be used for a meaningful analysis of literary hypotheses.

In this paper, we (1) introduce an annotated sentiment dataset of historical literary Danish and Norwegian, (2) evaluate various classification models on the dataset, and (3) use an accurate classifier to analyze a large historical literary corpus and provide initial evidence for a literary hypothesis as outlined in the following section. Our code and data are available at https://github.com/mime-memo/unhappy.

2 Literary Context: The Unhappy Texts

In our experiments, we focus on literature from the so-called 'Modern Breakthrough,' which denotes the literary currents embedded in the new social realist and naturalist literature which blossomed in Scandinavia during the 1870s and the 1880s as well as a cultural and societal transformation, encompassing politics, morals, gender roles, etc (Ahlström, 1947). On the whole, the historiography of the breakthrough, largely concentrated on a few canonical male authors, does not reflect the diversity of the period's literary production, especially in terms of gender. Women were most noticeable among the new groups of authors stepping

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 324–334

forward from 1870-1900. As a rule, female authors were not recognized by contemporary (male) criticism; the famous critic Georg Brandes introduced the term breakthrough in a book grouping only male Scandinavian authors, *The Men of the Modern Breakthrough* (Brandes, 1883). and, a generation later, the influential literary historian Vilhelm Andersen dubbed the period's female literary production as an 'odd province' of literature (Andersen, 1925). While Andersen recognized the period's literary production by Danish female authors, a much later literary handbook *Hovedsporet* (Jørgensen, 2005) does not. Its chapter on the debates on gender in the period's literature has the heading 'Women's literature, written by men'.

In fact, for many decades of the 20th century, the female breakthrough authors were overlooked and mostly left out of literary histories, but in the 1970's and 80's, as a part of a greater feminist wave in Scandinavian literary studies, scholars made a great effort to reinterpret the period. A result of this was the prevalent hypothesis about the works of the female authorships from the period being characterized as 'unhappy texts' (Dalager and Mai, 1982; Jensen et al., 1993). The argument is unfolded in greatest nuance in Pil Dahlerup's doctoral dissertation, *The women of the modern breakthrough* (Dahlerup, 1984). Dahlerup argues that because the women of the period were restrained by the patriarchal society, they also wrote fictional characters who lacked agency, were unhappy and disillusioned. Meanwhile the scholars highlight that the emancipated female characters of the period are first and foremost present in the works of male authors, arguing that the male authors, who enjoyed both private and public liberty, had the capacity to portray fictional characters with the same liberties (Dahlerup, 1984).

Our investigation will revisit this hypothesis with two points of departure, one regarding empirical selection and scale, and another regarding methodology. The hypothesis of the unhappy texts is based solely on texts by female authors and often on relatively few texts. The first point of departure is thus to revisit the hypothesis with a quantitative perspective, also including the works authored by men. Our second point of departure, in revisiting the hypothesis of the unhappy texts, is based on a theoretical framework from the field of gender studies and feminist literary studies. Central to our inquiry are new insights from affect theory and the notion of sentiment and affects as something humans 'do' rather than an underlying *a priori* structure as is the case in the psychoanalytic assumptions that the hypothesis of the unhappy texts is based on. We thus set out to find new methodologies to test the hypothesis of unhappy texts. In this experimental methodological phase, we find that sentiment analysis, analyzing the valence associated with a given word, sentence, or text, is a meaningful starting point because of its juxtaposition of affect theoretical and quantitative perspectives.

3 Related Work

In this section, we present sentiment-related works in both literature and historical corpora.

3.1 Sentiment Analysis on Literature

The study of emotions in literature has become an integral part of literary analysis with the emergence of digital humanities. This field of research focuses on using computational methods to understand emotions in literature. This encompasses a wide range of topics, from tracking changes in the plot to analyzing the emotional content of texts (Kim and Klinger, 2019).

Emotion and sentiment classification involves classifying text into predefined classes based on emotions/sentiment. This task is applied in the literature to group literary texts based on their emotional properties. Some studies have focused on classifying the emotions in works of Francisco de Quevedo's poems (Barros et al., 2013), American poetry (Reed, 2018), and early American novels (Yu, 2008). Volkova et al. (2010) annotated in fairy tales, while Ashok et al. (2013) used sentiment polarity to predict the success of a book. Zehe et al. (2016) used sentiment classification to classify 212 German novels into happy or non-happy endings. They showed encouraging results using support vector machines. While some studies approach the task as a classification problem, others focus on the structural changes of sentiment and emotions. Heuser et al. (2016) explore the relationship between emotions and geographical locations and Taboada et al. (2006) track sentiment and emotions towards certain groups.

3.2 Sentiment Analysis on Historical Text

Identifying and tracking the sentiment of text over time is challenging due to language variation, the

dynamic nature of sentiment, and the scarcity of historical corpora. Sentiment can change depending on a variety of factors, such as context, culture, and time. Additionally, historical and diachronic data may have different characteristics than contemporary data, such as changes in language use and writing styles. Along with the above challenges, the main focus of research nowadays is on temporal corpora in the news (Souma et al., 2019), and social media (Hazimeh et al., 2019), while little attention has been devoted to the historical domain (Sprugnoli et al., 2016).

Several techniques are used for sentiment analysis on diachronic and historical corpora. Schmidt and Burghardt (2018) used sentiment analysis on a German drama text corpus and evaluated the performance of different German sentiment lexicons using a manually annotated gold standard of 200 speeches. This study created an annotated corpus for the sentiment analysis of historical texts and revealed key issues related to the annotation and pre-processing of historical texts. Sprugnoli et al. (2016) analyzed an Italian corpus of writings of Alcide De Gasperi and developed a new lexical resource for sentiment analysis. The study found that crowd-sourcing was more effective for sentiment analysis of historical texts than using a sentiment lexicon. Hills et al. (2019) analyzed national subjective well-being using millions of digitized books in six different languages and countries. They found that Gross Domestic Product (GDP) and life expectancy have a strong positive effect on well-being, while conflict has a negative effect. These studies demonstrate the challenges and potential of sentiment analysis of historical texts and the importance of manual annotation and crowd-sourcing for accurate analysis.

Schmidt et al. (2021) analyzed emotional expression in historical German plays using various methods, including lexicon-based, traditional machine learning, word embeddings, and pre-trained and fine-tuned language models. The latter achieved state-of-the-art results, while lexicon-based and traditional machine learning consistently outperformed. However, performance decreases significantly with multiple sub-emotions.

To summarize, the challenges in sentiment analysis of historical texts, including lack of native speakers, limited data, unusual textual genres, and historical language, call for innovative and robust methods (Sprugnoli, 2021). Pre-trained language models can address these challenges by being trained on large corpora, fine-tuned on small datasets, and able to identify sentiment in diverse texts (§5).

	Main Corpus	**Sub-corpus**
Total novels	839	
Total sentences	3,229,137	2,748
Total words	52,724,457	55,333
Average sentences per novel	3,849	
Average words per novel	62,842	
Average words per sentence	16	20

Table 1: Corpus statistics for the main corpus and sub-corpus used in sentiment analysis.

4 Dataset of Historical Literary Text

This section describes our main corpus of historical literary text and a sub-corpus annotated for sentiment.

4.1 Main Corpus

We rely on the MEMO corpus (Bjerring-Hansen et al., 2022), comprising 839 Danish and Norwegian novels spanning the last 30 years of the 19th century and including more than 50 million words in total. We refer to this corpus as the 'main corpus'. The corpus is a rich and diverse collection of texts that will provide valuable insights into the registered sentiments and emotions of the period under investigation. Table 1 shows statistical information about the corpus.

4.2 Annotated Sub-Corpus

To ensure the accuracy and reliability of our sentiment classifiers, we systematically annotated a representative subset of sentences from our main corpus. We carefully selected 2,748 sentences, averaging 3.3 sentences per novel, to create a diverse sample. To minimize bias, we employed random sampling by shuffling the sentences and selecting 3-4 sentences from each novel. This method ensured that the annotations accurately reflect the sentiment distribution in the entirety of the corpus and laid a strong foundation for further research in sentiment analysis of historical novels. Additionally, Table 1 provides further statistical information regarding the sub-corpus.

4.3 Annotation Process

The annotation was conducted by three trained literary scholars: a master's student, a Ph.D. stu-

dent, and an associate professor. All three are native Danish speakers and annotated 690, 1029, and 1029 text segments from the corpus, respectively. The annotators shared domain knowledge in 19th century Scandinavian literature, with overarching and intertwining areas of expertise, but also particular interests (gender, history of ideas, and cultural history).

Guidelines. With respect to the principle that clear and simple instructions are crucial for obtaining high-quality annotations (Mohammad, 2016), and acknowledging the array of intricacies and bafflements, which an emotional analysis of literary texts based on small fragmentary segments raises, the guidelines were rigorously minimalist and pragmatic. Our guidelines are a simple sentiment annotation questionnaire with clarifying annotation directions.

1. The text segments were to be labeled either 'Positive', 'Negative', or 'Neutral'. The annotator was to assess which of these labels was most descriptive of the overall sentiment expressed in the segment.

2. Only the segment in question should be considered. Contextualisation and 'guessing' on what might go on before or after the segment were ruled out. In cases of doubt, the label should be 'Neutral'.

3. Attention should be paid to the historical fluctuations in language and semantic change. As an example the adjective 'besynderlig', today means 'weird', while it in the 19th century also had the meaning 'special' or 'curious'. This means that the following sample should be labeled as positive, rather than, and contrary to an anachronistic reading, negative:

   > 'Hun blev strax noget fortumlet over disse uforberedte Kjærtegn ; men da hun laa i hans Arme , saa' hun op paa ham med et besynderligt, ikke fornærmet Blik' (English translation: 'She was at once somewhat taken aback by these unprepared caresses; but as she lay in his arms, she looked up at him with a curious, not offended, look')

4. Finally, pragmatism was to be deployed by the annotators. Since the segments are short and heterogeneous (containing both dialogue, non-dialogue/description, or a mixture), no extra weight could be given to particular word classes (such as verbs, adjectives, or nouns).

Challenges. Since sentiment analysis has traditionally been used on texts with a strong valence and subjectivity (e.g. reviews or tweets), literary texts such as novels pose a challenge because they often are characterized by ambiguity and can be understood and interpreted in several different ways. Therefore, a moderate inter-annotator agreement (IAA) is also to be expected (Schmidt et al., 2019).

Another fundamental challenge in the annotation of imaginative texts reflects a key question in narratology (i.e., the study of narrative structure in texts): Who is speaking? In this context, distinctions are made between the text's (implied) author, narrator, and characters as well as between 'mood' and 'voice' as levels in literary texts (Genette, 1983). Since the narratological structure is often unclear, the annotation cannot aim at deciding whether the dialogue or text voices reflect the author's sentiment. At the risk of not taking into account literary devices such as irony and unreliability, the annotation can only address the dominant sentiment in the segment.

In addition, particular attention was paid to the following issues:

Lack of context. Often, segments with clear indications of emotion are ambiguous or vague due to a lack of context. We decided to label such instances as 'Neutral', even though they in reality, might have formed parts of negative vis-à-vis positive discourses. Example (translated): 'When the letter, finally, was finished, she folded it and went with a beating heart to the nobleman's door'. Here the valence is clear (something emotional is going on), but the polarity is unclear (is it something good or bad?).

(Unconscious) Contextualization. We tried our best to rely on our judgments on close reading combined with our familiarity with historical modes of morality and sensibility, including standards of courtesy and decorum, as expressed in speech as well as gesture and action. However, it is difficult not to contextualize or to rule out the role of (unconscious) contextualization. For instance, our knowledge of individual authors or texts and their distinctive traits (e.g., the Norwegian author Amalie Skram, famous for her coarse and unsentimental naturalistic style, permeated with negativity) or paratextual effects such as the connotations

of a book title (e.g., the novel *En Krise* (1892), 'A Crisis' by Johanne Schjørring, inevitably giving the reader a negative vibe towards the text).

Gender. Finally, and crucially in this context, is the fact that cultural values and norms change over time. Awareness of the issue of cultural value change over time, affecting the annotation (gender roles, gendered behavior, then and now). Relying on fundamental insights from social constructivist gender theory, understanding gender as a category subject to change in form and meaning over time, special attention was paid to segments involving gendered behavior and dialogue. A few examples can illustrate how cultural change poses challenges to annotation. First, we have a segment highlighting a male character, whose dominant behavior would be interpreted more negatively today than in the past:

> 'aa ja , men ofte gjorde han sig med Vilje haard , ti saaledes, var det bedst for dem, han havde at gøre med: "du skal!"' (Translation: 'oh yes, but often he made himself tough on purpose, for this was the best for those he had to deal with: "you must!"')

In the thematization of female gender roles, a modern understanding could potentially come at odds with the intentions in 19th century texts. This is highlighted in the following segment presenting a female character:

> 'Men allerhelst laa hun dog i Vinteraftenernes Skumring i sin Yndlingsstilling i Armstolen og grublede og drømte og ventede og ventede — ligesom Prinsessen i Eventyret.' (Translation: 'But preferably she would lie in the twilight of the winter evenings in her favourite position in the armchair and pondering and dreaming and waiting and waiting — like the princess in the fairy-tale.'

Today, the character's passivity, dreamfulness, and nostalgia are more likely to be perceived as negatives than positives.

4.4 Annotation Results

The sentiment annotations for all sentiment classes are illustrated in Table 2, which displays the distribution of the samples and sentiment classes.

Agreement. We use Cohen's Kappa to determine IAA on 100 samples annotated by all three experts, resulting in a score of 0.59, which indicates moderate agreement among annotators. This is likely a result of the subjectivity of the task and the challenges encountered in determining sentiment with limited context (see Challenges above).

Sentiment Class	Total Samples	Percentage
Negative	1,139	41.4%
Neutral	788	28.7%
Positive	821	29.9%
Total	2,748	100%

Table 2: Distribution of sub-corpus samples and sentiment categories.

5 Experiments

The dataset is split into three sets: training, validation, and testing to facilitate the development and evaluation of our models. The training set comprises 2376 examples, accounting for approximately 86% of the dataset. The validation set used for hyperparameter selection consists of 272 samples, representing about 10% of the entire dataset. Lastly, the testing set, which is utilized to evaluate the final performance of the model, contains 100 examples, representing approximately 4% of the total dataset. In the case of the training and validation sets, annotations were performed by a single expert. However, for the testing set, samples were annotated by all three experts, and the final label was determined through a majority vote. We use F1-score as our evaluation metric.

5.1 Lexicons and Models

In this section, we outline the lexicons and models evaluated in our sentiment classification experiments using both lexical-based, supervised without fine-tuning, and supervised with fine-tuning methods. Importantly, all lexicons and models are based on modern Danish lexica and/or training data, with no exposure to historical Danish or Norwegian. It should be noted that, until 1907, written Norwegian was practically identical to written Danish (Vikør, 2022). In section 5.2, 5.3, and 5.4, we show detailed information on how these resources were employed to achieve sentiment classification results. The following provides a concise overview of the lexicons and BERT models employed in our experiments.

Sentida. A Danish lexicon[1] comprised of the existing Danish sentiment lexicon AFINN and a list of new words. It scores sentences based on their words and provides either an average sentiment

[1] https://github.com/Guscode/Sentida

score or a total score. Sentida takes into account adverb-modifiers, exclamation marks, and negations in its sentiment scoring process (Lauridsen et al., 2019).

TextBlob. A Python package[2] that utilizes a lexicon-based approach in which sentiment is determined by the semantic orientation and the strength of each word in the sentence, using a pre-existing English dictionary that categorizes words as negative, neutral, or positive. We employ a two-step process to ensure accurate sentiment in Danish text. First, we utilize Google Translate[3] to translate Danish text to English. Then, we feed the translated text into TextBlob.

VADER. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool attuned explicitly to sentiments expressed in social media and works well on texts from other domains (Hutto and Gilbert, 2014).[4] To adjust the VADER sentiment analysis technique for Danish, we use a Danish sentiment lexicon (Nimb et al., 2022; Pedersen et al., 2021)[5] containing a list of words and their associated sentiment scores.

Danish Model BotXO. This BERT model was developed by Certainly (previously BotXO) (Devlin et al., 2019). It has been pre-trained on 1.6 billion Danish words and is freely available[6]. This particular model has not been fine-tuned on sentiment classification.

Danish BERT Tone. The BERT Tone model[7] was developed to detect sentiment polarity (positive, neutral or negative) in Danish texts. The model was constructed by fine-tuning the BotXO Danish BERT model.

Danish Sentiment. This model[8] is a fine-tuned version of the multilingual pre-trained model XLM-RoBERTa-base (Conneau et al., 2020). It has been fine-tuned on 198M tweets specifically for sentiment analysis.

[2] https://textblob.readthedocs.io/

[3] https://translate.google.com/

[4] https://github.com/cjhutto/vaderSentiment

[5] https://github.com/dsldk/danish-sentiment-lexicon

[6] https://huggingface.co/Maltehb/danish-bert-botxo

[7] https://huggingface.co/alexandrainst/da-sentiment-base

[8] https://huggingface.co/vesteinn/danish_sentiment

Lexicon/BERT Model	Valid.	Test
Lexicon-based		
Sentida	0.64	0.63
TextBlob	0.56	0.52
VADER	0.59	0.62
Supervised (without fine-tuning)		
Danish BERT Tone	0.59	0.62
Danish Sentiment	0.71	0.74
Supervised (with fine-tuning)		
Danish BERT BotXO	0.50	0.52
Danish BERT Tone	0.59	0.70
Danish Sentiment	0.63	0.72

Table 3: Lexicon-based and supervised sentiment classification F1-Score using different methods on validation and testing sets.

5.2 Lexicon-based classification experiment

Lexicon-based sentiment analysis is a technique used to determine the sentiment of a given text by assigning positive, negative, or neutral values to individual words based on their meanings. This approach relies on a pre-built sentiment lexicon, which contains a list of words and their corresponding sentiment values. The sentiment of a text is then calculated by summing the sentiment values of the individual words within the text. We evaluate the performance of our sentiment classifiers on the validation and testing split of the sub-corpus dataset on the Sentida, TextBlob, and VADER lexicons.

The Sentida classifier achieves the best results with 64% and 63% on the validation and testing sets, respectively. The results of the other two lexicons are presented in Table 3. Further analysis reveals that Sentida is the best at transferring to the unseen domain and language variation of historical literary Danish and Norwegian.

5.3 Supervised classification experiments (without fine-tuning)

To predict sentiment from a pre-trained model without fine-tuning, we first load the pre-trained model, format the input data and make predictions. However, without fine-tuning the model to rec-

ognize patterns specific to a particular sentiment analysis task, its performance may be limited.

We evaluate the performance of two pre-trained Danish BERT models, the Danish BERT Tone, and Danish Sentiment. The results show that the Danish Sentiment overpasses the Danish BERT Tone in both, evaluation and testing sets.

5.4 Supervised classification experiments (with fine-tuning)

The dataset used in lexicon-based and supervised without fine-tuning classification is also utilized in conducting supervised classification experiments. We fine-tune and evaluate three different pre-trained language models (BERT BotXO, Danish BERT Tone, and Danish Sentiment) in our dataset.

In this experiment, we fine-tune the described pre-trained BERT models on a task-specific dataset for sentiment classification. The dataset consists of 2,748 labeled sentences, with an almost balanced distribution of positive, neutral, and negative sentiments. We use a batch size of 32 and train the model for 30 epochs, using the AdamW optimizer with a learning rate of 10^{-3} (Loshchilov and Hutter, 2017). We evaluate the performance of each model using F1-score. Here we observe larger differences between validation and testing, which is a result of the fact that model selection (number of training epochs) was performed using the validation set. The Danish Sentiment model achieved the highest F1-score of 63% and 72% on the validation and test sets, respectively. Table 3 shows details about the obtained results from each model.

6 Classifier-assisted Corpus Analysis

We employ the 'Danish Sentiment' BERT model, which has shown to be the top-performing model, for predicting the sentiment of all sentences in the main corpus. We align the sentiment with the author's gender and the novel's year of publication. In Figure 1, the distribution of sentiment levels is depicted in relation to the author's gender and the percentage of sentences. Notably, female authors tend to exhibit a higher proportion of both negative and positive sentiments compared to male authors, on average.

These results provide insights into potential differences in sentiment expression between male and female authors in the analyzed data. The literary hypothesis about female authorships being

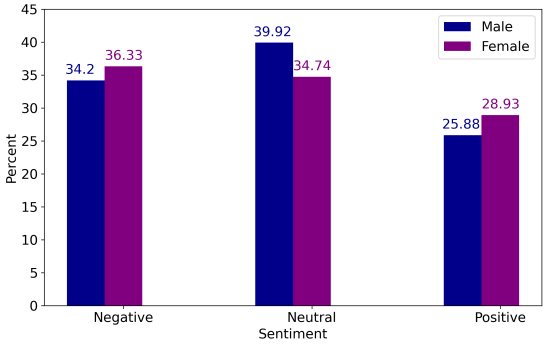


Figure 1: Distribution of sentiment and the author's genders. The X-axis is the sentiment class. Y-axis is the percentage of sentences per sentiment category.

'unhappy' is partly confirmed by our analysis. The female authors did express more negative sentiment, but also more positive sentiment than the male authors. Thus the implied positivity of the male authors, which is also part of the literary hypothesis, is not confirmed. A preliminary new hypothesis could therefore be that the female authors of the period wrote with a more expressed sentiment, whilst the male authors had a tendency towards a more unaffective style of writing.

Figure 2 provides a detailed analysis of the sentiment distribution over time concerning the author's gender. This figure shows the changes in sentiment tendencies across different gender groups and how these sentiments evolve over a given period. The results provide valuable insights into the dynamic nature of sentiments based on gender and can aid in understanding the evolving nature of sentiments and gender influences. The figure demonstrates that female authors exhibit more negative and positive sentiments over time than male authors.

Additionally, we calculated the overall sentiment by using a weighted average approach that considered both the sentiment distribution and weights of positive, neutral, and negative sentiments. Specifically, we assigned a weight of 3 to negative sentiment, 2 to neutral sentiment, and 1 to positive sentiment. To compare the sentiment of male and female authors over time, we computed the average sentiment scores over a period of 29 years and plotted them in Figure 3. Higher scores indicated more negative sentiment, while lower scores indicated more neutral or positive sentiment. The results in the figure show that, on

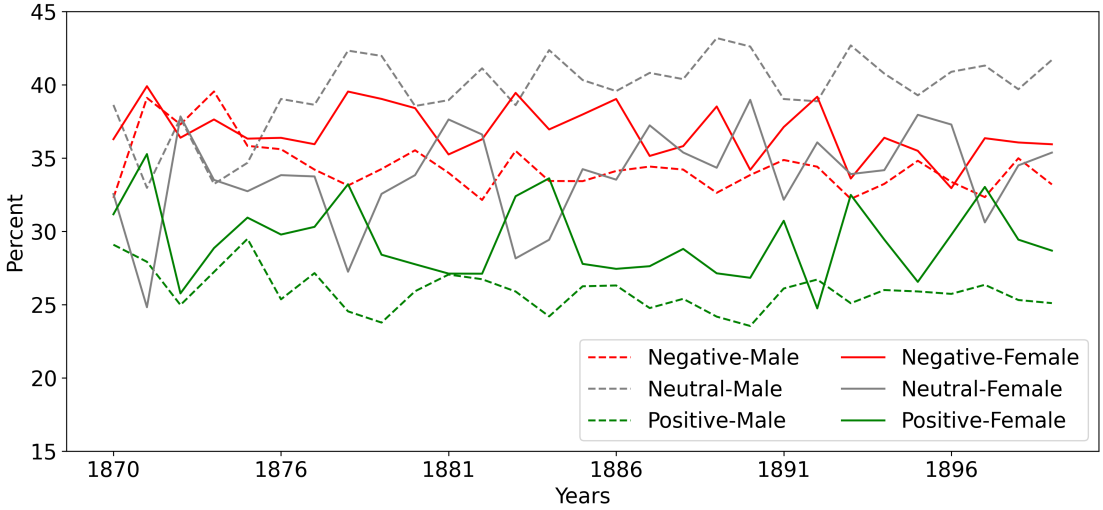


Figure 2: Distribution of sentiment over time. The X-axis is the years. The Y-axis is the percentage of sentences of a particular sentiment out of all sentences by authors of the same gender from that year.

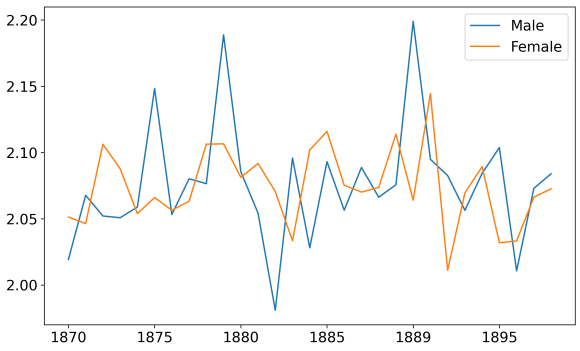


Figure 3: Average sentiment for both male and female authors over time.

Sentiment Class	p-value
Negative	5.20×10^{-6}
Neutral	5.13×10^{-8}
Positive	4.57×10^{-10}

Table 4: Significance of sentiment class differences: p-values from t-tests comparing mean male and female sentiment groups over the years, revealing statistically significant differences in sentiment trends.

average, the negative sentiment scores of female authors are higher than those of male authors in 16 out of the 29 years analyzed.

The significance of our findings was assessed by performing a t-test for each sentiment category, with the null hypothesis that the mean number of sentences with that category across the years is the same for sentences by male and female authors. In statistical analysis, a p-value lower than 0.05 is generally considered statistically significant and enables us to reject the null hypothesis, indicating the counts are statistically different. Table 4 shows small, statistically significant p-values for all sentiment classes.

7 Discussion

These initial analyses of a literary corpus with the newly developed sentiment classifier both confirm and contest the thesis of the 'unhappy' female texts. On the face of it, it seems that the female authors in the corpus express not only negative but also positive sentiments. At the same time, the second (implicit) part of the thesis is refuted, as the male authors do not write more positively. Clearly, this is reopening the literary discussion of the unhappy text with a level of complexity that these results cannot account for alone.

Further work is needed. In terms of computational interventions, a more in-depth analysis would call for us to investigate polarity in greater detail (how positive or negative are the novels?), while also paying more attention to narrative structures (how does polarity relate to

plot?).

More trivial analytical next steps involve employing segmentation of data and comparisons of sub-corpora. Are there, for example, certain works or authors that are outliers in terms of gender or valence? Are there any correlations when comparing canonized works and popular literature? By zooming in on either part of the novels (the endings, for example) or parts of the corpus (with regards to time slots or subgroups of producers/individual authors), specific strong trends may stand out.

Also, by working with smaller subsets of the corpus, we could perhaps challenge the essentialist idea of an 'Écriture Féminine', a unique feminine style of writing. Indeed, instead of simply asking whether there is a difference between male and female authorship in the period, we would also be able to explore how this difference is created in the corpus and possibly cultural, historical, gendered, and narratological reasons for it. In this connection, the fact that male authors seemingly write more 'neutrally' while female authors write with a higher degree of valence might be put into perspective through Sara Ahmed's theory of the 'stickiness' of emotions. Inspired by performativity theory, Sara Ahmed is interested in what emotions or affects do, rather than what they are (Ahmed, 2004). Both 'positive' affects, such as (naïve) joy, and 'negative' affects, such as shame have, especially during the period in question but also still today, clung to the feminine, while hegemonic masculinity has been and is associated with the rational and unsentimental (Connell, 1995).

In other words, a further refinement of the analytical steps goes hand in hand with critical interactions with the theoretical framework and specific historical contexts concerning the concepts of gender and emotion.

8 Conclusion

In this work, we used the MEMO corpus to create a high-quality human-annotated sentiment dataset for historical literary Danish and Norwegian. Despite multiple challenges, we showed that the task is feasible and that inter-annotator agreement is sufficiently high to warrant the use of the dataset for the training and evaluation of sentiment classifiers. In an extensive evaluation of such models, we found that the best performance is obtained with XLM-RoBERTa fine-tuned on sentiment analysis of modern Danish tweets and then further on the training set from our dataset. Using this model to annotate the whole MEMO corpus automatically, we observe that, as the literary theory predicts, female authors expressed more negative sentiments than male authors in the period. However, the hypothesis of the unhappy texts is only partially confirmed by our analysis as the numbers also suggest that the female authors, in contrary to the hypothesis, expressed more positive sentiments than the male authors who collectively expressed more neutral sentiments than their female counterparts.

In future work, we intend to experiment with further improvements of the classifier, including pre-training a language model on the whole corpus as a basis for fine-tuning the sentiment dataset. Furthermore, we intend to complement the classifier with topic models to investigate the evolution of sentiment towards specific topics (cultural change) over time and the evolution of words used to express a sentiment (language change).

From a literary perspective, a sentiment classifier trained on Scandinavian novels from the end of the 19th century holds great potential outside of the context of our specific test case. To give just one example, it would be interesting to study the so-called 'naturalism' from the time of the Modern Breakthrough. This literary movement or current is often said to be characterized by a pronounced pessimistic worldview. One can assume that negative segments predominate naturalist writings, and if that is in fact the case, the numbers would be interesting to compare with the rest of the literature from the period, i.e., canonical, realistic novels and this period's many forgotten texts.

From a literary theoretical perspective, a classifier such as this can be used to shed light on the importance of narratological layers at a macro level. Although the human reader is far superior to algorithms in separating character, narrator, and (implied) author, the numerical output of the computer is very interesting for existing research in the period. Questions like 'Does the sentiment distribution correspond to the overall assertions of the individual works?' and 'How does sentimental quantity relate to literary expression?' become possible to investigate.

References

Gunnar Ahlström. 1947. *Det moderna genombrottet i Nordens litteratur*. Kooperativa förbundets bokförlag.

Sara Ahmed. 2004. *The cultural politics of emotion*. Edinburgh University Press.

Vilhelm Andersen. 1925. *Den danske litteratur i det nittende aarhundredes: Illustreret dansk litteraturhistorie*. Gyldendal.

Vikas Ganjigunte Ashok, Song Feng, and Yejin Choi. 2013. Success with style: Using writing style to predict the success of novels. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1753–1764.

Linda Barros, Pilar Rodriguez, and Alvaro Ortigosa. 2013. Automatic classification of literature pieces by emotion detection: A study on Quevedo's poetry. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 141–146. IEEE.

Jens Bjerring-Hansen, Ross Deans Kristensen-McLachlan, Philip Diderichsen, and Dorte Haltrup Hansen. 2022. Mending fractured texts. a heuristic procedure for correcting OCR data.

Georg Brandes. 1883. *Det moderne gjennembruds maend*. Gyldendal.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

R. W. Connell. 1995. *Masculinities*. Polity Press.

Pil. Dahlerup. 1984. *Det moderne gennembruds kvinder.*, 2. edition. Gyldendal.

Stig Dalager and Anne-Marie Mai. 1982. *Danske kvindelige forfattere*, volume 2. Gyldendal.

G. Genette. 1983. *Narrative Discourse: An Essay in Method*. Cornell UP.

Hussein Hazimeh, Mohammad Harissa, Elena Mugellini, and Omar Abou Khaled. 2019. Temporal sentiment tracking and analysis on large-scale social events. In *Proceedings of the 2019 8th International Conference on Software and Computer Applications*, pages 17–21.

Ryan Heuser, Mark Algee-Hewitt, and Annalise Lockhart. 2016. Mapping the emotions of london in fiction, 1700–1900: A crowdsourcing experiment. In *Literary mapping in the digital age*, pages 43–64. Routledge.

Thomas T Hills, Eugenio Proto, Daniel Sgroi, and Chanuki Illushka Seresinhe. 2019. Historical analysis of national subjective wellbeing using millions of digitized books. *Nature human behaviour*, 3(12):1271–1275.

Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225.

Elisabeth Møller Jensen, Eva Hættner Aurelius, Inger-Lise Hjordt-Vetlesen, Margaretha Fahlgren, Unni Langås, Lone Finnichand, and Anne-Marie Mai. 1993. *Nordisk kvindelitteraturhistorie*, volume 2. Rosinante.

Jens Anker Jørgensen. 2005. *Hovedsporet: Dansk litteraturs historie*. Gyldendal A/S.

Evgeny Kim and Roman Klinger. 2019. A survey on sentiment and emotion analysis for computational literary studies. *Zeitschrift für digitale Geisteswissenschaften*.

Gustav Aarup Lauridsen, Jacob Aarup Dalsgaard, and Lars Kjartan Bacher Svendsen. 2019. SENTIDA: A new tool for sentiment analysis in Danish. *Journal of Language Works-Sprogvidenskabeligt Studentertidsskrift*, 4(1):38–53.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Saif Mohammad. 2016. A practical guide to sentiment annotation: Challenges and solutions. In *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 174–179.

Sanni Nimb, Sussi Olsen, Bolette Sandford Pedersen, and Thomas Troelsgård. 2022. A thesaurus-based sentiment lexicon for Danish: The Danish sentiment lexicon. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2826–2832.

Bolette Sandford Pedersen, Sanni Nimb, and Sussi Olsen. 2021. Dansk betydningsinventar i et datalingvistisk perspektiv. *Danske Studier*.

Ethan Reed. 2018. Measured unrest in the poetry of the black arts movement. In *DH*, page 477.

Thomas Schmidt and Manuel Burghardt. 2018. An evaluation of lexicon-based sentiment analysis techniques for the plays of Gotthold Ephraim Lessing. Association for Computational Linguistics.

Thomas Schmidt, Manuel Burghardt, Katrin Dennerlein, and Christian Wolff. 2019. Sentiment annotation for Lessing's plays: Towards a language resource for sentiment analysis on German literary texts. In *2nd Conference on Language, Data and Knowledge*, pages 45–50.

Thomas Schmidt, Katrin Dennerlein, and Christian Wolff. 2021. Using deep learning for emotion analysis of 18th and 19th century German plays.

Wataru Souma, Irena Vodenska, and Hideaki Aoyama. 2019. Enhanced news sentiment analysis using deep learning methods. *Journal of Computational Social Science*, 2(1):33–46.

Rachele Sprugnoli. 2021. Sentiment analysis for Latin: a journey from Seneca to Dante Alighieri. *Sentiment Analysis in Literary Studies*.

Rachele Sprugnoli, Sara Tonelli, Alessandro Marchetti, and Giovanni Moretti. 2016. Towards sentiment analysis for historical texts. *Digital Scholarship in the Humanities*, 31(4):762–772.

Maite Taboada, Mary Ann Gillies, and Paul McFetridge. 2006. Sentiment classification techniques for tracking literary reputation. In *LREC workshop: towards computational models of literary analysis*, pages 36–43.

Mads Rosendahl Thomsen, Emma Risgaard Olsen, Telma Peura, and Mads Nansen Paulsen. 2021. Desire, digital literary studies – a companion guide. In *https://litdh.au.dk/topics/desire*.

Lars S. Vikør. 2022. Rettskrivingsreform i store norske leksikon på snl.no. In *https://snl.no/rettskrivingsreform*.

Ekaterina P Volkova, Betty Mohler, Detmar Meurers, Dale Gerdemann, and Heinrich H Bülthoff. 2010. Emotional perception of fairy tales: achieving agreement in emotion annotation of text. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 98–106.

Bei Yu. 2008. An evaluation of text classification methods for literary study. *Literary and Linguistic Computing*, 23(3):327–343.

Albin Zehe, Martin Becker, Lena Hettinger, Andreas Hotho, Isabella Reger, and Fotis Jannidis. 2016. Prediction of happy endings in German novels based on sentiment information. In *3rd Workshop on Interactions between Data Mining and Natural Language Processing, Riva del Garda, Italy*, pages 9–16.

Parser Evaluation for Analyzing Swedish 19th-20th Century Literature

Sara Stymne[1], **Carin Östman**[2], **and David Håkansson**[2]
[1]Department of Linguistics and Philology, Uppsala University, Sweden
[2]Department of Scandinavian Languages, Uppsala University, Sweden
sara.stymne@lingfil.uu.se
{carin.ostman,david.hakansson}@nordiska.uu.se

Abstract

In this study, we aim to find a parser for accurately identifying different types of subordinate clauses, and related phenomena, in 19th–20th-century Swedish literature. Since no test set is available for parsing from this time period, we propose a lightweight annotation scheme for annotating a single relation of interest per sentence. We train a variety of parsers for Swedish and compare evaluations on standard modern test sets and our targeted test set. We find clear trends in which parser types perform best on the standard test sets, but that performance is considerably more varied on the targeted test set. We believe that our proposed annotation scheme can be useful for complementing standard evaluations, with a low annotation effort.

1 Introduction

Dependency parsers can be useful tools for analyzing large text materials, and as such can enable large-scale studies within many scientific disciplines. Modern parsers can achieve very high scores on standard test sets, at least for languages with large treebanks, but these test sets are often limited to only a few domains, and typically to publication-level modern language, such as news or Wikipedia. For more challenging text types, for instance, noisy data like Twitter or historical texts, parsers typically perform considerably worse even for high-resource languages.

Parsers are typically evaluated on a treebank that is split into training, development, and test sets. This can overestimate the parser performance, since parsers are then trained on data that matches its test set in all relevant aspects, such as genre, time period, and annotation style. Furthermore, parser evaluation is typically done using metrics that give a holistic score for the full tree, such as (un)labeled attachment score. In many real-world scenarios, such as ours, we are not interested in the full tree, but in a subset of relations.

This study is part of a larger project with the overall aim to identify and explore language change in Swedish literature during the period 1800–1930. During the 19th century, the Swedish language changed in several aspects. This change includes various linguistic levels and also involves lexical aspects. Overall, the changes led to a smaller difference between spoken and written Swedish since the written language moved closer to the spoken vernacular (see Section 3). The goal of the project is to cover morphological, syntactical, and lexical changes. In this paper, however, we focus only on syntactic aspects, focusing on subordinate clauses. The changes in the 19th century resulted in a less complex language — not least as far as subordinate clauses and related phenomena are concerned. To enable large-scale analysis of subordinate clauses, we require a high-quality parser for our target domain, Swedish literary novels and short stories from 1800–1930. In this paper, we explore whether parsers can be evaluated for this domain, without requiring a large manual annotation effort.

To evaluate a parser for a new text type and task, as in our case 19th century literature with a focus mainly on subordinate clauses, we would ideally like to have an annotated treebank for the target text type. However, this is a human annotation task that is time-consuming, and thus costly, and which requires an expert on dependency grammar. For many practical projects, this is not feasible. We propose a lightweight annotation task for our target task, which consists of only annotating one type of phenomenon per sentence. The focus is on four phenomena related to subordinate clauses, for which we annotate a small targeted test set for our target text type. For comparison, we also evaluate

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 335–346

Relation	Example	Translation	Class
CLEFT	Det var här han skulle *anfallas* .	'It was here that he would be *attacked* .'	Correct
CLEFT	Det skola vi *göra* klockan åtta .	'That we should *do* at eight o'clock'	Wrong
RELCL	Hvad hon beundrar Mauritz , som kan *stå* så lugn !	'How she admires Mauritz , who can *stand* so calmly !'	Correct
RELCL	Men kan du säga hvar vi *äro* ?	'But can you tell me where we *are* ?'	Wrong
CCOMP	Se till att du inte *halkar* .	'Make sure that you do not *slip* .'	Correct
CCOMP	Må den aldrig mer *komma* för mina ögon !	'May it never again *come* before my eyes !'	Wrong
NO-AUX	Jag har fått hvad du i natt *skrifvit* till mig .	'I have received what you [have] *written* for me tonight .'	Correct
NO-AUX	Enhälligt ha vi *kommit* fram till detta slut :	'Unanimously , we have *reached* this end :'	Wrong

Table 1: Examples of sentences shown to the annotators, marked as either correct or wrong.

on standard Swedish test sets. Table 1 shows examples of each class, where the task is to identify if a given word is the head of a specific subordinate clause type or if it is a clausal complement without the auxiliary 'have'.

We compare several variants of three generations of parsers trained on different subsets of the Universal Dependencies (UD) treebanks (Nivre et al., 2020), and evaluate them on UD, both with holistic metrics and for a subset of relations of interest, as well as on our targeted test set. On the UD test sets we see clear trends that a modern transformer-based parser is better than BiLSTM- and SVM-based parsers, and that it is better to train on several North Germanic languages than only on Swedish. However, on our new targeted test set, the results are more mixed, and we see less clear trends, which is in line with earlier work for German (Adelmann et al., 2018). We think that our targeted test set is able to give a complementary view to standard evaluations, but that the sampling procedure can be improved.

In Section 2 we review related work, followed by a description of our project focused on Swedish language change in Section 3. In Section 4 we describe the data and in Section 5 we describe the parsers evaluated, including the multilingual training setup. We present the results in Section 6, discuss them in Section 7, and finally, we conclude in Section 8.

2 Related Work

Dependency parsers have continuously developed, from 'old school' parsers like MaltParser (Nivre et al., 2007) and MSTparser (McDonald et al., 2005) based on classical machine learning, like support vector machines, to modern neural parsers. Many of the first strong neural parsers were based on recurrent neural networks, as most of the best parsers in the CoNLL 2017 shared task on dependency parsing (Zeman et al., 2017). Next, models based on deep contextualized embeddings have been taking over, and most strong parsers today are based on fine-tuning contextualized models like BERT (Devlin et al., 2019) or XLM-R (Conneau et al., 2020), e.g. Machamp (van der Goot et al., 2021) and Trankit (Nguyen et al., 2021).

The standard way to evaluate dependency parsers is by calculating holistic metrics such as labeled attachment score (LAS), which measures the percentage of words which gets both their head word and label correct. There are, however, examples of more detailed evaluations (e.g. McDonald and Nivre, 2007; Kulmizev et al., 2019; Salomoni, 2017), focusing on aspects such as arc and sentence lengths, non-projective dependencies, and scores for specific POS-tags and dependency relations. The overall conclusion is typically that different parser types have different strengths, e.g. that graph-based parsers tend to perform better than transition-based parsers on long-distance dependencies (McDonald and Nivre, 2007). As far as we are aware, there are no datasets and evaluations like our proposal, focused on a single relation per sentence.

Highly relevant to our study is the work of Adelmann et al. (2018), who evaluate a set of six parsers for digital humanities research, focusing on German novels and academic texts. Like us, they are also interested in specific relations, for instance, related to speaker attribution, and not only in holistic evaluation. Unlike us, they perform a full dependency tree annotation effort for three sample texts. In addition, they do not include any neural parsers in their evaluation. They find that several parsers do well on the holistic metrics, but that the results are considerably worse for several of the specific relations of interest, such as appositions, and that it is not always the overall strongest

parser that is the best choice for a specific relation. Salomoni (2017) performed a detailed evaluation on parsing German 17th-century literature, for which he annotated two excerpts of text with full dependency annotations. Again, no neural parsers were included in the study, which found a drop compared to in-domain results, but where the relative performance of the two parsers evaluated was consistent on different metrics, possibly because of the large difference in performance between them.

Swedish literary texts from different eras have been analyzed for different purposes before, requiring taggers and/or parsers. Dahllöf (2022) aims to characterize differences between dialogue and narrative in contemporary fiction, whereas Stymne et al. (2018b) analyze prose rhythm in a novel from 1940. However, in none of these studies, the choice of tagger and/or parser is motivated. There have also been some earlier smaller-scale studies focusing on the transition towards a more colloquial written Swedish. For instance, language development in Swedish literature during the 19th century has been explored, but only on a small scale focusing on individual authors (e.g. Lindstedt, 1922; Von Hofsten, 1935).

3 Language Change in 19th Century Swedish

This study is part of a larger project with the overall aim to identify and explore language change in Swedish literature during the period 1800–1930. In the history of the Swedish language, this period is characterized by modernization in the sense that the written language was influenced by the spoken vernacular. In this process of modernization, fictional prose is of certain interest since it has been suggested that linguistic change spread from literary dialogue (Engdahl, 1962; Teleman, 2003). By investigating a corpus of literary texts the project will not only contribute with a more detailed account of language change in 19th-century Swedish but also address the question of how linguistic change increased in the community.

The modernization of the Swedish written language during the 19th century affected several linguistic aspects. As for the lexicon, it is well-known that formal function words were replaced by colloquial counterparts. Much attention has also been devoted to the loss of verbal agreement, i.e. the use of the vernacular singular variant in both singular and plural. On the syntactic level, Engdahl (1962) has shown a remarkable change in sentence length during the end of the 19th century. Engdahl's study focuses on non-fictional prose, periodicals from 1878 to 1950, but his results call for a more detailed account of syntactic complexity during the period, and hence we will focus on subordinate clauses and phenomena related to them in this paper.

For this study, we have chosen to focus on three types of subordinate clauses, based on UD dependency labels, and one phenomenon related to subordinate clauses: (i) *relative clauses* (RELCL), (ii) *cleft constructions* (CLEFT),[1] (iii) *clausal complements* not determined by obligatory control (CCOMP), and (iv) *auxiliary drop* (NO-AUX). Whereas the first three types can be used in order to measure syntactic complexity, auxiliary drop has been suggested to mark written style, and hence almost never occur in spoken language (cf. Wellander, 1939). Since auxiliary drop of finite verbs is restricted to subordinate clauses in Swedish, we have included it as related to subordinate clauses. In this study, we only include auxiliary drop that occurs in clausal complements, CCOMP. Examples of the selected clause types are shown in Table 1.

4 Data

In this section, we will describe the existing data from UD and the new targeted dataset we constructed for this project

4.1 Universal Dependencies Treebanks

We use data from Universal Dependencies (Nivre et al., 2020) version 2.11 (Zeman et al., 2022) for training our parsers and for the standard evaluation. Besides dependency annotations, UD also contains lemmas, universal and language-specific part-of-speech tags (UPOS/XPOS), and morphological features. Our main focus is on Swedish, for which there are three treebanks, Talbanken, LinES, and PUD, where PUD only contains a test set. In addition, we use data from related North Germanic languages: Norwegian (both variants: Bokmål and Nynorsk), Danish, Faroese, and Icelandic. The treebanks used are summarized in Table 2. The intuition behind also using related lan-

[1] In UD, both relative clauses and cleft constructions are subtypes of ACL, clausal modifier of noun, and are denoted ACL:RELCL and ACL:CLEFT. In this paper, we will use shorter names, excluding the prefix.

Language	Treebank	Code	Genres	Train	Test
Swedish	Talbanken	sv_t	news, nonfiction	67K	20K
	PUD	sv_p	news, wiki	–	19K
	LinES-M	sv_lm	fiction, nonfiction, spoken	18K	73K
Norwegian	Bokmaal	no_b	blog, news, nonfiction	244K	30K
	Nynorsk	no_n	blog, news, nonfiction	245K	25K
	NynorskLIA	no_nl	spoken	35K	10K
Danish	DDT	da	fiction, news, nonfiction, spoken	80K	10K
Faroese	FarPaHC	fo	bible	1.5K	6.6K
Icelandic	Modern	is	news, nonfiction	7.5K	10K

Table 2: Treebanks used, with info about genres (as defined in UD) and the number of tokens in test and training data. LinES-M refers to our modified version of LinES.

guages is twofold, first, it has been shown to improve parsers (e.g. Smith et al., 2018a), second, we believe it may make the parser more robust to non-standard Swedish, which has many differences from the modern Swedish of the Swedish treebanks. Written Norwegian and Danish, in particular, are very similar to Swedish, and are considered mutually intelligible.

As can be seen in Table 2, the genres, according to the UD specification, of the treebanks used are mixed. To be able to, at least some extent, investigate whether it would help to have an in-genre test set, we create a modified version, LinES-M, of the LinES treebank (Ahrenberg, 2007) which consists of three genres: literary fiction, Microsoft manuals, and European parliament proceedings. The literary part contains a set of novels translated from English, published 1977–2017. While this is not a perfect match to our target of novels and short stories written originally in Swedish during an earlier time period, this was the closest we could get to an in-domain test set, without any re-annotation. We re-split LinES by merging the data from the training and test sets, and moving all literature[2] to a new test set, and all other texts to a new training set, referred to as LinES-M in Table 2.

For evaluation on the UD test sets, we report labeled attachment score (LAS). For LinES-M, we also report F1-scores for the three relations in focus for our targeted test set and AUX, which is relevant for identifying auxiliary drop.

4.2 Targeted Literature Dataset

In this section, we will describe the sampling and annotation of the targeted literary dataset annotated for this project as an alternative way of evaluating the performance of parsers on specific phenomena in a specific text type. The targeted dataset is publicly available under the Creative Commons license, CC BY-NC-SA.[3]

4.2.1 Text Selection and Processing

Our target data is literary texts from 1800–1930, focusing on novels and collections of short stories. Such works have been made available by Litteraturbanken.[4] We choose to work only with the subset of works that have been proofread after going through OCR, available in an XML format. We extracted all novels and short stories available in this format from the time period of interest. From these texts, we extracted the raw text paragraphs, and tokenized the text. For another sub-project, we had already extracted a set of novels where quotation marks are used to mark dialogue, and used the quotation marks to separate dialogue and narrative, which we use also in this study. This sample consists of 165 novels and collections of short stories. The data was parsed early on in the project, using Swepipe and UUparser[s] with Swepipe tags (see Section 5).

The annotation task was designed to be simple and fast. Thus we decided to focus on a single relation of interest per sentence. From the parse trees, we extracted all sentences containing an arc labeled with a relation of interest and marked the modifier of the arc, which is the headword of the specific subordinate clause.[5] Figure 1 shows a parsed example sentence, containing a relative clause, with an arc from the headword *Mauritz* to the modifier *stå* ('stand'), which is the head of the

[2]The literary works are in documents 2,3,4,6,7, and 8; document 1 contains Microsoft manuals and document 5 contains parliament proceedings (Lars Ahrenberg, personal communication).

[3]https://github.com/UppsalaNLP/SweSubEval

[4]https://litteraturbanken.se/

[5]It would also be possible to consider other more complex annotations, such as also including the head of the relevant arc, to ensure that the subordinate clause is attached at the correct position, or to require that the span of the subordinate clause is correct.

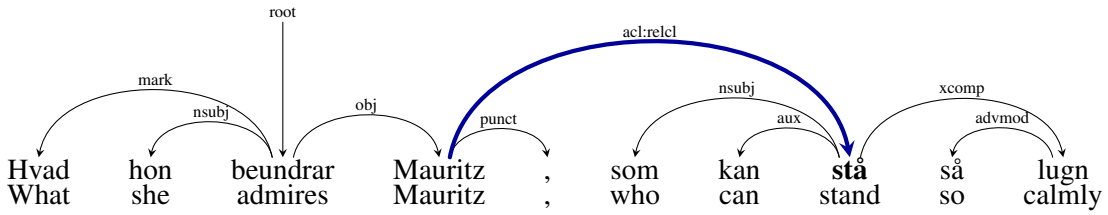


Figure 1: Parsed dependency tree (UUParser) for the sentence *Hvad hon beundrar Mauritz, som kan *stå* så lugn* 'How she admires Mauritz, who can *stand* so calmly', with English glosses added. The arc of interest, ACL:RELCL with 'stand' as a modifier, is marked in blue.

relative clause *som kan stå så lugn* ('who can stand so calmly'), where the marked word thus would be *stå*, as an instance of the type RELCL. For NO-AUX, we also checked that there was no outgoing AUX relation from the marked word. It is not uncommon to have several instances of a single relation type in a sentence, but we only marked a single occurrence per example, to make the annotation consistent between sentences. From this set, we randomly sampled 200 sentences for each relation type, except CLEFT, for which we only found 74 examples, which were all included. Table 1 shows annotated examples of each class, where we also see examples of old plural verb forms like *äro* (modern: *är*, 'are') and old-fashioned spelling like *'skrifvit'* (modern: *skrivit*, 'written').

4.2.2 Annotation

The annotation was performed by the last two authors, both native Swedish speakers, and researchers in Scandinavian languages with expertise in Swedish grammar. The annotators were given the example sentences in Excel, and for each sentence, they were to decide whether the marked head word belonged to the given type or not. For each type, 20 examples were annotated by both annotators, and the remaining examples were split between them. After the first round, there were a few disagreements in the doubly annotated sets, which were discussed by the annotators, followed by a re-annotation of all examples. The initial round of annotation was very quick, roughly between 15–30 minutes per 100 examples, with a somewhat longer time needed for CCOMP. Table 3 shows the number of correct and wrong examples for each class. Note that the dataset is skewed towards positive examples.

4.2.3 Evaluation

We evaluate on the targeted dataset by calculating the number of times the parser assigns the correct relation to the focus word, and for NO-AUX, that there in addition is no aux-dependent. We then calculate precision and recall for each relation type. Note that recall may be overestimated by this procedure since we do not cover any examples not identified by a parser. This evaluation is different from standard evaluation of dependency parsers where we evaluate a full tree. In this case, we instead evaluate a single relation of interest for each sentence.

Relation	Correct	Wrong
CLEFT	64	10
RELCL	133	67
CCOMP	141	59
NO-AUX	170	30

Table 3: Class distribution in our annotated dataset

5 Parsers

In order to investigate how well the different types of evaluation work, we explore three generations of parsers. As a baseline, we use the easily accessible Swepipe with its provided model for Swedish. We also use two generations of neural parsers, UUParser and Machamp, for which we also experiment with multilingual parsing. We train each model three times with different random seeds and report average scores.

5.1 Swepipe

As a baseline parser, we wanted an easily accessible parser, which comes with a trained parsing model, and which might be used by non-experts in a digital humanities project. Our choice was to use the Swedish annotation pipeline, Swepipe.[6], a pre-trained model covering all steps needed to analyze Swedish texts from scratch, including tokenization, tagging, and parsing. Swepipe is similar

[6] https://github.com/robertostling/efselab

to several other systems targeted at this user group, such as the web-based Swegram,[7] which uses the same parser and tagger (Megyesi et al., 2019).

Swepipe is pre-neural and uses efselab (Östling, 2018) for tagging and MaltParser (Nivre et al., 2007) trained on Talbanken for parsing. MaltParser is a classical transition-based parser, using a support vector machine for classification, based on a feature vector with words, POS-tags, and already built relations.

5.2 UUParser

UUParser (de Lhoneux et al., 2017; Smith et al., 2018b) is a neural transition-based dependency parser with a BiLSTM feature extractor, based on Kiperwasser and Goldberg (2016). Word representations are fed to a BiLSTM, to create contextualized word representations, which are given as input to an MLP classifying the next transition. We use an arc-hybrid transition model (Kuhlmann et al., 2011) with a swap transition (Nivre, 2009) and a static-dynamic oracle (de Lhoneux et al., 2017). As input word representation we use word embeddings, character-based word embeddings, UPOS-tag embeddings, and treebank embeddings, which represent the treebank of a sentence. All embeddings were initialized randomly at training time. When applying UUparser on new texts, we need a proxy treebank that indicates which of the treebanks from training for use as the treebank embedding at test time, for which we always use Talbanken, since it is present in all models, and it performed well in Stymne et al. (2018a). We use the default UUparser settings (Smith et al., 2018b), except for adding drop-out with a rate of 0.33 for UPOS-embeddings, since the parser is trained with gold tags. At test time, we use two different sets of POS-tags, from Swepipe/efselab and from Machamp. We will call these variants UUparser^s and UUparser^m respectively. To counteract the differing sizes of the training data, we limited the number of sentences used per treebank to 4,300 per epoch.

5.3 Machamp

Machamp (van der Goot et al., 2021) is a toolkit for multitask learning covering several NLP tasks, based on fine-tuning a pre-trained contextualized model, like BERT (Devlin et al., 2019). In a multi-task setup, each task has a separate decoder. The dependency parser is a graph-based parser using deep biaffine attention (Dozat and Manning, 2018) to score word pairs, and the CLU algorithm (Chu and Liu, 1965; Edmonds, 1967) to extract trees. For tagging, a greedy decoder, with a softmax output layer is used.

In this work we use Machamp in a multi-task setup, to jointly learn tagging of UPOS, XPOS, and morphological features, and dependency parsing. We experiment with two sets of language models, multilingual BERT (mBERT Devlin et al., 2019),[8] trained on 104 languages including all languages used in our study except Faroese, and the Swedish model KB-BERT (Malmsten et al., 2020), trained only on Swedish. We will call these systems Machamp^m and Machamp^k respectively. For both models, we used the cased version.[9] KB-BERT has been shown to improve Swedish named entity recognition and POS-tagging (Malmsten et al., 2020), but as far as we are aware, it has not been used in multilingual dependency parsing models. We use the default parameters of Machamp. To counteract the differing sizes of the training data, we applied sampling smoothing set to 0.5.

Group	Included treebanks/languages
Talbank	Swedish-Talbanken
Swedish	Talbank+ Swedish-LinES-M
SweNor	Swedish + Norwegian (*3)
Scand	SweNor + Danish
NorthG	Scand + Faroese + Icelandic

Table 4: Groups of languages/treebanks used for multilingual training. See Table 2 for specific treebanks.

5.4 Multilingual Training

For UUParser and Machamp, we explore multilingual training. We limit ourselves to the North-Germanic languages, all relatively closely related to Swedish. We train two Swedish models, on Talbanken only, to be comparable with Swepipe, and also with LinES-M. In addition, we train three models with different subsets of the other North Germanic languages. For our multilingual models, we first combine Swedish with Norwegian, which has three treebanks covering both variants

[7] https://cl.lingfil.uu.se/swegram/

[8] https://github.com/google-research/bert/blob/master/multilingual.md

[9] We used models from HuggingFace (https://huggingface.co/models), for KB-BERT: KB/bert-base-swedish-cased and for mBERT: bert-base-multilingual-cased.

	LAS			F1, LinES-M			
	LinES-M	TB	PUD	CLEFT	RELCL	CCOMP	AUX
Swepipe-Talbank	71.75	79.69	78.82	–	61.31	54.98	88.45
UUparser^m-Talbank	72.10	83.75	76.66	26.82	64.67	59.62	93.99
UUparser^m-Swedish	75.51	83.76	77.50	29.12	67.37	61.65	94.21
UUparser^m-Norswe	79.69	85.60	81.50	39.92	74.34	66.79	94.35
UUparser^m-Scand	79.74	85.43	81.34	41.74	73.03	64.93	94.20
UUparser^m-NorthG	79.33	85.35	81.27	41.71	72.82	64.70	94.27
Machamp^k-Talbank	80.54	**92.24**	86.05	**56.73**	79.07	**74.59**	**95.44**
Machamp^k-Swedish	80.26	90.72	**86.83**	49.67	75.84	71.29	93.94
Machamp^k-Norswe	**83.13**	91.63	86.79	55.42	**81.29**	**75.32**	95.29
Machamp^k-Scand	**83.16**	**92.31**	**87.21**	**55.54**	**81.21**	74.27	**95.97**
Machamp^k-NorthG	**83.03**	**92.35**	**87.17**	**56.00**	**82.27**	**74.78**	**95.85**

Table 5: Results on standard Swedish UD test sets. LAS for all three Swedish test sets, and F1-scores for four relations of interest for LinES-M.

	Precision				Recall			
	CLEFT	RELCL	CCOMP	NO-AUX	CLEFT	RELCL	CCOMP	NO-AUX
Swepipe-Talbank	–	66.33	70.41	84.62	0.00	**99.25**	**98.57**	**97.06**
UUparser^m-Talbank	92.46	93.32	94.11	98.14	50.35	82.37	63.97	51.44
UUparser^m-Swedish	92.49	93.45	95.84	97.60	69.79	81.45	65.95	50.85
UUparser^m-NorSwe	92.12	94.65	**97.39**	98.30	**84.55**	81.20	70.87	56.21
UUparser^m-Scand	94.64	**95.69**	**96.73**	98.72	**84.20**	79.62	70.48	61.05
UUparser^m-NorthG	93.31	95.55	**96.06**	**99.05**	**75.00**	79.37	74.13	61.57
Machamp^k-Talbank	94.12	95.16	94.63	98.52	59.90	83.46	**75.48**	**65.69**
Machamp^k-Swedish	94.92	**96.19**	95.09	98.81	53.12	82.21	**73.81**	65.10
Machamp^k-NorSwe	**95.38**	**96.71**	94.77	**99.13**	72.92	79.70	73.33	**67.25**
Machamp^k-Scand	**96.61**	95.11	94.29	**99.01**	59.38	**87.47**	66.90	58.82
Machamp^k-NorthG	**95.38**	93.83	93.46	99.00	64.06	**87.72**	68.10	58.04

Table 6: Precision and recall for our targeted test set.

of Norwegian. We then add Danish, to train a Scandinavian model. The reason for adding Norwegian first, despite the fact that Danish is considered a closer relative to Swedish, is the availability of more data for Norwegian with variability in language variants. Our final model, NorthG, also adds Faroese and Icelandic, which are more distant from Swedish, and not mutually intelligible. The language groups are summarized in Table 4.

6 Results

Tables 5 and 6 show results from the standard and targeted evaluations for Swepipe, UUparser^m with Machamp^k POS-tags and Machamp^k trained with KB-BERT. In all tables, we mark the three best results for each metric in bold. While our focus is on Swedish, which is reported in this section, we also report results with Machamp for the additional languages used for training our parsing models in Appendix A.

Table 5 shows results on UD test sets. We see no obvious differences between the LAS performance pattern on the in-genre LinES-M and the other two Swedish test sets, indicating that genre may not play a big role in this case; contemporary novels are likely relatively close to the news, non-fiction, and wiki texts in the other Swedish treebanks. Swepipe has overall the lowest scores, followed by UUparser^m, and then Machamp^k. For the two Swedish models, the differences between using only Talbanken and adding the small LinES-M training set are typically small, but sometimes with a positive effect for UUparser^m and a negative effect for Machamp^k.[10] Adding Norwegian leads to improvements in nearly all scores, often quite substantial, whereas adding additional languages has a smaller impact. The difference between parsers varies for the different relation types. Swepipe does not find any CLEFTs, and falls behind UUparser^m on all other relation types, especially for AUX. Machamp^k improves considerably over UUparser^m for all explored relations, except AUX, where both neural parsers perform well, possibly since they both use the POS-tags of Machamp^k.

[10]This may be due to the fact that in Machamp, treebanks are simply concatenated, but in UUparser, they are distinguished by treebank embeddings, which has been shown to improve results when training on different treebanks for the same language (Stymne et al., 2018a). We leave an investigation of this issue to future work.

	LAS			F1, UD_LinES-M				P, litt			
	LinES-M	TB	PUD	CLEFT	RELCL	CCOMP	AUX	CLEFT	RELCL	CCOMP	NO-AUX
Swepipe-Talbank	71.75	79.69	78.82	–	61.31	54.98	88.45	–	79.52	82.14	90.41
UUparser^s-Talbank	70.80	82.35	75.78	26.08	63.01	58.39	91.31	92.80	92.52	93.05	96.50
UUparser^s-Scand	77.63	83.39	80.25	30.77	70.55	62.22	90.82	93.86	94.07	**94.66**	97.95
UUparser^m-Talbank	72.10	83.75	76.66	26.82	64.67	59.62	**93.99**	92.46	93.32	94.11	98.14
UUparser^m-Scand	79.74	85.43	81.34	41.74	73.03	64.93	94.20	**94.64**	**95.69**	**96.73**	98.72
Machamp^m-Talbank	77.20	89.35	84.21	38.47	72.87	69.09	92.91	92.94	**96.13**	93.00	98.23
Machamp^m-Scand	**80.13**	**89.50**	**85.79**	**43.09**	**77.67**	**71.18**	93.49	93.41	**96.98**	92.47	**99.08**
Machamp^k-Talbank	**80.54**	**92.24**	**86.05**	**56.73**	**79.07**	**74.59**	**95.44**	**94.12**	95.16	**94.63**	**98.52**
Machamp^k-Scand	**83.16**	**92.31**	**87.21**	**55.54**	**81.21**	**74.27**	**95.97**	**96.61**	95.11	94.29	**99.01**

Table 7: Comparison of parser variants, on standard test sets and our test set.

The results in Table 6 for our targeted test set show a partially different picture. First, we note that Swepipe has a very high recall for all relation types except CLEFT, which it never predicts. We think this is mainly an artifact of the sampling procedure for this test set, where the annotated sentences were sampled from Swepipe and UUparser^s, with Swepipe POS-tags, which means that they were mostly predicted as correct by Swepipe. The other parsers do not have this advantage and thus have a lower recall, which we believe is more predictive of real performance, even though it still may be overestimated due to the sampling procedure. Swepipe has considerably lower precision than the other parsers for all relation types. We believe that the evaluation should still be fair in comparing UUparser^m and Machamp^k, from which no samples were taken. Compared to the standard evaluation where Machamp^k was clearly better than UUparser^m, we now see a more mixed picture, where there is no clear overall advantage of Machamp^k over UUparser^m, and the results are mixed across relation types and precision/recall. The trends between training languages are also less clear, with some combinations standing out in performance for some relation types. Machamp^k trained with Scand and NorthG has a considerably higher recall on RELCL than the other models, with only a small drop in precision. On CCOMP and NO-AUX, on the other hand, these two models instead have a low recall, without gaining much on precision. We do not see this pattern for UUparser^m, where the Scand model is overall strong.

In Table 7 we show a summary of results for both variants of UUparser and Machamp, showing only precision for the targeted test set, since recall is biased towards Swepipe and UUparser^s due to the sampling.[11] We can see that UUparser^s does not consistently improve on LAS over Swepipe when trained on the same Talbanken data, but that adding the Scandinavian treebanks improves the results considerably both for the UD evaluations and on the targeted test set. When we compare the two variants of UUparser and Machamp we see that UUparser^m and Machamp^k beat their variant consistently on the UD evaluation, and in most cases on the targeted test set. We also see that training on Scand is better than training on Talbanken in the majority of cases, both for UD and on precision for the targeted test set, however, from Table 6, we know that Scand is sometimes not as strong on recall.

7 Discussion

An important question is whether the parser performance on our target task is good enough to use for our study of change in the Swedish written language. Overall, both Machamp and UUparser have good precision for all our relations of interest, always scoring above 90, and reaching scores above 96 for some parsers for each relation type. The recall, however, is considerably lower. This means that the instances of each relation type the parser finds are mostly good, but it does miss a substantial part of relevant instances, especially given the fact that all examples are sampled from a parser, and we might have missed additional instances. The recall is highest for RELCL, where it is well above 80 for several of the models both for Machamp and UUparser. This approaches a level that is usable for our end project, of finding syntactic features in 18th–19th-century literature, and tracking them over time. Other relation types have a more mixed performance, as CLEFT, for which UUparser^m trained on NorSwe and Scand performs very well, with a recall of over

[11] To save space, we only show results for two training language groups. The other groups exhibit largely the same trends.

84, but where other models perform considerably worse. The recall of CCOMP, and especially of NO-AUX is lower, and we would need to improve parser performance for those relation types, possibly by using domain adaptation techniques, before they would reach a useful level. The varying performance of parsers for different relation types is in line with the results for German by Adelmann et al. (2018), who recommend choosing different parsers for different end goals.

On the standard evaluation, Machamp is clearly overall better than UUparser, training on Scand is better than training only on Swedish, KB-BERT is better than mBERT for Machamp, and UUparser is better with Machamp tags than with Swepipe tags. For our targeted test sets, however, we see fewer clear trends, and there is much more variation among the systems. Machamp^k and UUparser^m tend to perform better than their counterparts, and the multilingual models may have a small advantage over the Swedish-only models. Swepipe clearly seems to fall behind the other parsers on precision, whereas its high recall can be explained by the sampling procedure. A side-effect of our study is that we have found that Machamp^k trained on Scand or NorthG is a very strong parser for modern Swedish as measured by the UD test sets.

Our targeted test set does suffer from an issue with sampling from only two parsers, which affects its recall mainly for Swepipe, but also for UUparser^s. We believe UUparser^m is less affected since it relies on a different set of POS-tags. The dataset is also relatively small, especially for the CLEFT relation. However, we think it still contributes to showing that when selecting a parser for a particular target task and text type, we cannot rely solely on evaluation scores on standard test sets, as also shown by Adelmann et al. (2018). Even if we focus on the F1-score for the relations of interest in Lines-M, rather than on the full tree, we see no clear similarity of parser ranking to the evaluation of the same relation types in our targeted test set. To further investigate whether this type of test set can indeed be useful, we would need to perform further analysis. It would be interesting to learn more about where the main improvements shown on UD evaluation for a parser like Machamp^k actually occurs. We also think it would be useful to consider the sampling for the test set, specifically to also annotate some raw text, in order to find out what type of instances are not identified by any of our parsers. Another issue that we did not yet explore, is whether parsing performance varies over the time period in question.

8 Conclusion

We describe a study of Swedish dependency parsers with the goal of tracking changes in the use of certain types of subordinate clauses and related phenomena in Swedish literature from 1800–1930. Since standard test sets do not cover this time period or genre, and we did not have the resources to perform a full annotation of dependency trees, we propose a smaller-scale annotation task, focusing on single relation types. We evaluated a set of parsers on UD and on our targeted test set. While there was a clear and relatively consistent order between the parsers on the UD evaluation, the performance was more mixed on our targeted test set, without a clear overall best parser across relation types. We believe that our proposed annotation scheme can be useful in complementing standard evaluations, with a low annotation effort, but that more analysis is needed.

Acknowledgments

This work is funded by the Swedish research council under project 2020-02617: *Fictional prose and language change. The role of colloquialization in the history of Swedish 1830–1930*. We would like to thank Johan Svedjedal and Joakim Nivre for their helpful discussions about this work, and the anonymous reviewers for their insightful comments. Computations were enabled by resources in project UPPMAX 2020/2-2 at the Uppsala Multidisciplinary Center for Advanced Computational Science.

References

Benedikt Adelmann, Wolfgang Menzel, Melanie Andresen, and Heike Zinsmeister. 2018. Evaluation of out-of-domain dependency parsing for its application in a digital humanities project. In *Proceedings of the 14th Conference on Natural Language Processing (KONVENS 2018)*, pages 121–135, Vienna, Austria.

Lars Ahrenberg. 2007. LinES: An English-Swedish parallel treebank. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, pages 270–273, Tartu, Estonia. University of Tartu, Estonia.

Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Mats Dahllöf. 2022. Quotation and narration in contemporary popular fiction in swedish – stylometric explorations. In *Proceedings of the 6th Digital Humanities in the Nordic and Baltic Countries Conference (DHNB 2022)*, pages 203–211, Uppsala, Sweden.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240.

Sven Engdahl. 1962. *Studier i nusvensk sakprosa. Några utvecklingslinjer*. Skrifter utgivna av Institutionen för nordiska språk vid Uppsala universitet, Uppsala.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Marco Kuhlmann, Carlos Gómez-Rodríguez, and Giorgio Satta. 2011. Dynamic programming algorithms for transition-based dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 673–682, Portland, Oregon, USA. Association for Computational Linguistics.

Artur Kulmizev, Miryam de Lhoneux, Johannes Gontrum, Elena Fano, and Joakim Nivre. 2019. Deep contextualized word embeddings in transition-based and graph-based dependency parsing - a tale of two parsers revisited. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2755–2768, Hong Kong, China. Association for Computational Linguistics.

Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2017. Arc-hybrid non-projective dependency parsing with a static-dynamic oracle. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 99–104, Pisa, Italy. Association for Computational Linguistics.

Torvald Lindstedt. 1922. Studier över stilen i Gösta Berlings saga. *Nysvenska studier*, 2:31–77.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the National Library of Sweden - making a Swedish BERT. *CoRR*, abs/2007.01658.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 122–131, Prague, Czech Republic. Association for Computational Linguistics.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Beáta Megyesi, Anne Palmér, and Näsman Jesper. 2019. SWEGRAM – Annotering och analys av svenska texter. Technical report, Department of Linguistics and Philology, Uppsala University.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90, Online. Association for Computational Linguistics.

Joakim Nivre. 2009. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 351–359, Suntec, Singapore. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryiğit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering, 13(2),*, 13(2):95–135.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Robert Östling. 2018. Part of speech tagging: Shallow or deep learning? *Northern European Journal of Language Technology*, 5:1–15.

Alessio Salomoni. 2017. Dependency parsing on late-18th-century German aesthetic writings: A preliminary inquiry into Schiller and F. Schlegel. In *Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage*, DATeCH2017, page 47–52, New York, NY, USA. Association for Computing Machinery.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018a. 82 treebanks, 34 models: Universal Dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Aaron Smith, Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2018b. An investigation of the interactions between pre-trained word embeddings, character models and POS tags in dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2711–2720, Brussels, Belgium. Association for Computational Linguistics.

Sara Stymne, Miryam de Lhoneux, Aaron Smith, and Joakim Nivre. 2018a. Parser training with heterogeneous treebanks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 619–625, Melbourne, Australia. Association for Computational Linguistics.

Sara Stymne, Johan Svedjedal, and Carin Östman. 2018b. Språklig rytm i skönlitterär prosa. En fallstudie i Karin Boyes *Kallocain*. *Samlaren. Tidskrift för forskning om svensk och annan nordisk litteratur*, 139:128–161.

Ulf Teleman. 2003. *Tradis och funkis : svensk språkvård och språkpolitik efter 1800*, 1st edition. Norstedts ordbok, Stockholm, Sweden.

Louise Von Hofsten. 1935. Några stildrag hos Selma Lagerlöf med utgångspunkt från Charlotte Löwenskiöld. *Nysvenska studier*, 15:150–183.

Erik Wellander. 1939. *Riktig svenska: en handledning i svenska språkets vård*. Norstedt, Stockholm, Sweden.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, et al. 2022. Universal dependencies 2.11. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

A Results for Additional Languages

Table 8 shows results for all treebanks used during training when parsed with Machamp either fine-tuned on top of the Swedish KB-BERT model (Malmsten et al., 2020), or the multilingual mBERT model (Devlin et al., 2019), when trained on the different language groups (see Table 4. We note that for Danish and Norwegian, which are very closely related to Swedish, models including these languages in the parsing training data perform nearly as well when trained on

		sv_t	sv_lm	sv_p	no_b	no_n	no_nl	da	fo	is
Talbank	KB-BERT	92.24	80.54	86.05	*72.33*	*64.33*	*45.62*	*62.61*	*18.56*	*11.00*
	mBERT	89.35	77.20	84.21	*77.93*	*75.10*	*51.89*	*67.73*	*37.15*	*48.31*
Swedish	KB-BERT	90.72	80.26	86.83	*71.78*	*63.84*	*46.03*	*62.14*	*17.03*	*9.76*
	mBERT	86.43	76.77	84.21	*77.82*	*74.68*	*51.87*	*67.71*	*35.53*	*46.62*
SweNor	KB-BERT	91.63	83.13	86.79	91.51	90.91	75.01	*69.15*	*20.81*	*13.85*
	mBERT	89.56	79.82	85.68	92.28	91.77	75.98	*72.25*	*37.08*	*50.38*
Scand	KB-BERT	92.31	83.16	87.21	91.73	91.36	75.57	86.44	*21.42*	*14.50*
	mBERT	89.50	80.13	85.79	92.01	91.63	75.41	87.13	*38.36*	*49.62*
NorthG	KB-BERT	92.35	83.03	87.17	91.94	91.47	75.68	86.54	63.49	31.94
	mBERT	89.49	79.99	85.77	92.15	91.49	75.94	87.24	71.82	60.30

Table 8: Results for all languages used for training models with Machamp fine-tuned based on Swedish KB-BERT or multilingual mBERT. Codes for treebanks refer to Table 2. Scores marked in italics indicate languages that were not present in the parser training data. All languages except Faroese are present in the mBERT pre-training data.

top of the Swedish KB-BERT model as on top of the mBERT model trained on 104 languages including Norwegian and Danish. It thus seems that for very similar languages a strong language model for a close language is just as good as a multilingual model containing many unrelated languages. However, this only holds when Danish and Norwegian are among the languages in the parsing training data; when the parser is trained only on Swedish, it is better to use mBERT than KB-BERT. Icelandic and Faroese are less closely related to Swedish than Danish and Norwegian, and for these languages, it is always better to use mBERT than KB-BERT. It is also notable that the performance is much poorer than for Danish and Norwegian. Faroese, which is not present in mBERT, performs quite poorly both with KB-BERT and mBERT when not present in the parser training data, but quite well with both models when present in the training data, whereas Icelandic even in that case performs poorly with KB-BERT. Overall, we see that Machamp with mBERT trained with the NorthG model is a strong parser for all the included languages. However, adding Icelandic and Faroese to the Scandinavian model has only a minor impact on the Scandinavian languages.

An Empirical Study of Multitask Learning to Improve Open Domain Dialogue Systems

Mehrdad Farahani[1] **Richard Johansson**[1,2]
[1]Chalmers University of Technology, [2]University of Gothenburg
{mehrdad.farahani, richajo}@chalmers.se

Abstract

Autoregressive models used to generate responses in open-domain dialogue systems often struggle to take long-term context into account and to maintain consistency over a dialogue. Previous research in open-domain dialogue generation has shown that the use of *auxiliary tasks* can introduce inductive biases that encourage the model to improve these qualities. However, most previous research has focused on encoder-only or encoder/decoder models, while the use of auxiliary tasks in *decoder-only* autoregressive models is under-explored. This paper describes an investigation where four different auxiliary tasks are added to small and medium-sized GPT-2 models fine-tuned on the PersonaChat and DailyDialog datasets. The results show that the introduction of the new auxiliary tasks leads to small but consistent improvement in evaluations of the investigated models.

1 Introduction

In recent years, open-domain dialogue systems have experienced increased research due to the availability of large corpora of dialogue and advances in deep learning techniques (Gao et al., 2018). Unlike task-oriented dialogue systems designed for specific domains or tasks, such as flight booking, hotel reservation, customer service, and technical support (Budzianowski and Vulić, 2019; Budzianowski et al., 2018; Chao and Lane, 2019), open-domain dialogue systems aim to have long-term connections with users by satisfying their emotional, social, and communication needs. Therefore, such a system must comprehend the dialogue context and user demands in order to select the appropriate skill at the appropriate time and generate consistent (Li et al., 2016b; Luan et al., 2017) and grounded (Ghazvininejad et al., 2018; Moon et al., 2019) interpersonal responses. Open-domain dialogue systems can include single-turn or multi-turn dialogues, where the context and topic of the conversation may change throughout the interaction (Dinan et al., 2020).

Multi-turn open-domain dialogue systems need to maintain the context of the conversation, generate appropriate responses concerning the context and predefined characteristics (persona), and handle various forms of input. A persona is a set of characteristics or attributes that describe a virtual agent's background, personality, and behavior. These attributes include the agent's name, age, gender, profession, interests, and other aspects (Zhang et al., 2018), while a conversation context refers to the background information or previous interactions relevant to the current conversation, with word-level and utterance-level dependencies (Zhao et al., 2020).

Recent developments in transformer-based architectures (Vaswani et al., 2017; Raffel et al., 2020; Lewis et al., 2020) for large-scale pre-training, such as OpenAI's GPT-2 (Radford et al., 2019), have shown exceptional results. Later, the models are fine-tuned via more technical steps and at different scales on a large-scale dialogue corpus (Zhang et al., 2020; Adiwardana et al., 2020; Thoppilan et al., 2022; Shuster et al., 2022).

Pre-trained models on conversational datasets typically process dialogue context (a list of utterances) as a sequence of tokens per utterance to generate responses. Although these approaches show effective results compared to previous approaches, they still need to catch the latent information in more complex structures rather than just tokens (Gu et al., 2021; Zhao et al., 2020). A conversational domain is distinguished by the presence of another component called utterances[1] that plays an imperative role in conveying higher-level information in addition to tokens and their local relationships. Recent research has put forth the use of *auxiliary tasks*

[1]An utterance is a spoken or written sentence or phrase that is used to convey meaning or participate in a dialogue.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 347–357

as a form of regularization during the fine-tuning of models as a means to address the aforementioned issue. A majority of these additional training objectives are implemented solely on the encoder-only and encoder-decoder architectures.

However, while the use of auxiliary tasks has led to improvement in encoder-only and encoder/decoder models, recent work has not explored the application of auxiliary tasks in *decoder-only* models. In this research, we propose incorporating auxiliary tasks on top of an autoregressive decoder-only model to examine and enhance the quality of generated responses concerning the latent information present within utterances. Additionally, we demonstrated the impact of various auxiliary tasks on distinct elements of dialogue across two benchmark datasets. By examining the effect of different auxiliary tasks on various components of dialogue, we aimed to provide a deeper understanding of how these tasks can influence the performance and outcomes of conversational systems. Additionally associated code to this research can be found in our GitHub repository.[2]

2 Related Works

The motivation for this research is drawn from recent investigations into the utilization of auxiliary tasks to enhance the generated responses in open-domain dialogue systems by considering the context. To this end, we present and analyze these recent studies in this section. Previous studies in this field can be broadly classified into three general categories. The first category pertains to the widespread use of encoder-decoder models in dialogue response generation, which have been observed to produce generic and uninteresting responses (e.g., "I'm good", "I don't know"). Zhao et al. (2020) proposed an encoder-decoder architecture with two auxiliary tasks at token and utterance levels that can effectively exploit conversation context to generate responses, including order recovery and masked context recovery. Analogously, Mehri et al. (2019) examined a range of unsupervised pre-training objectives for acquiring dialogue context representations via encoder-decoder models by incorporating four auxiliary tasks, including next-utterance retrieval, next-utterance generation, masked-utterance retrieval, and inconsistency identification.

DialogBERT is a unique design that employs a hierarchical Transformer architecture to comprehensively capture the context of dialogue (Gu et al., 2021). Using two training objectives, similar to BERT (Devlin et al., 2019), allows the model to understand a conversation's nuances effectively. In the first objective, masked context regression, the model is trained to predict the missing context from a dialogue, and in the second objective, distributed utterance order prediction, the model is trained to predict the order of spoken utterances in a conversation so that it understands the flow and logic.

Lastly, decoder-only models, like DialoGPT (Zhang et al., 2020), make use of only the final component of the encoder-decoder structure. DialoGPT in particular, extends the GPT-2 (Radford et al., 2019) architecture by being specifically developed and trained on a large corpus of dialogue data to generate responses in a conversational context. However, despite its ability to perform well in single-turn conversation, its lack of capability to capture latent information behind utterances in a multi-turn conversation, results in an inadequate understanding of the context. The utilization of auxiliary tasks in decoder-only models is a well-established practice. For instance, the GPT-2 based model TransferTransfo (Wolf et al., 2019), which adopts a multi-task objective, showed improvement over the basic GPT-2. These auxiliary tasks primarily take the form of sequence classification tasks.

3 Method

3.1 A Problem Definition

In this section, the necessary notations utilized are presented, and the learned tasks are briefly outlined. Let $d^{(i)} = (p_1, p_2, \ldots, p_N, u_1, u_2, \ldots, u_T)$ denote the i-th dialogue session in the dataset $\mathcal{D}$, where $\mathcal{C} = (u_1, u_2, \ldots, u_{T-1})$ is the dialogue context (history), $\mathcal{P} = (p_1, p_2, \ldots, p_N)$ is the dialogue persona (personality of the system) and u_T is the response regarding to the persona and the context. Each $u_i = \left(w_1^i, w_2^i, \ldots, w_{|u_i|}^i\right)$ in $\mathcal{C}$ is an utterance and w_j^i is the j-th word in u_i. Then, we aim to generate contextually relevant responses for multi-turn conversations using self-supervised auxiliary tasks. Our approach involves two major components, a language model trained based on the GPT-2 and a classification model on top of the GPT-2 used for auxiliary parts. This simple structure has been found to be effective in producing consistent responses. As such, two auxiliary

[2] https://github.com/m3hrdadfi/MoGPT

tasks have been designed over language modeling (LM) to improve the system's performance further. Order and masked recovery tasks are designed to enhance the self-attention module's capacity to capture linguistic affinities. The utterance permutation task enhances the self-attention module's ability to grasp word and utterance sequences, while the masking task seeks to reinforce semantic connections between words and utterances by optimizing the self-attention mechanism. These auxiliary tasks are critical in providing additional supervision signals to the model, leading to improved language modeling performance. Figure 1 illustrates the model. Lastly, a total loss function is defined to incorporate these auxiliary tasks and the primary objective of language modeling. It serves as the optimization target during training and guides the model toward producing accurate and consistent responses.

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \alpha \mathcal{L}_{\text{aux}} \quad (1)$$

Here, α is a hyper-parameter that controls the trade-off between LM and the objectives of the auxiliary tasks.

3.2 Auxiliary Tasks

Recent research (Sankar et al., 2019) has shown that Transformer-based autoregressive models are robust to unrealistic perturbations at both the utterance and word levels. However, despite this robustness, the study suggests that these models have learned a bag-of-words-like representation rather than genuinely understanding language structure and meaning. On the other hand, understanding context is crucial to producing coherent and consistent responses in open-domain dialogue systems. While the connections between words within an individual utterance are essential for determining the meaning, it is also necessary to consider the relationships between utterances to fully understand the context of the conversation. To enhance the language model's comprehension and its ability to generate accurate and consistent outputs, it was deemed necessary to provide additional means to understand the relationships between the order of utterances and their meaning and to capture the sequential structure of language, as well as to comprehend the relationships between individual words in an utterance to grasp the semantic structure of language. We propose two auxiliary tasks for this purpose in this paper.

3.2.1 Utterance Permutation (UP)

In order to retain the sequential structure of language, re-ordering utterances is defined as an auxiliary generator in two ways: detection or recovering methods by rearranging 10% of utterances in a dialog chosen by 15% of all dialogues in the collection. Depending on the dataset, this task can be implemented based on a persona, context (history), or both.

In this work, we considered two approaches to implementing UP as auxiliary tasks:

- *detecting* (UPD), implemented as a binary token classification task.
- *recovering* (UPR), implemented as a non-binary token classification task.

In UPR, we attempt to predict the correct tokens regarding the re-ordered tokens; in UPD, we only determine whether or not the tokens are in the right place.

3.2.2 Utterance Masking (UM)

In our effort to comprehend the semantic structure underlying utterances, we devised the utterance masking task. This task is executed using two distinct approaches, analogously to the two methods described above:

- *detecting* the tokens in the masked utterances (UMD), implemented as a binary classification task.
- *recovering* the tokens in the masked utterance (UMR).

In both methods, 15% of the tokens within each dialogue are selected, with 80% of these tokens being replaced in the non-binary approach by the <mask> token and by synonyms in the binary approach. In the non-binary method, 10% of the tokens were randomly substituted from the dictionary, while in the binary approach, they are replaced with antonyms. The final 10% of tokens were preserved in their original form.

4 Dataset and Experiments

The following section provides detail on the dataset and experimental settings used in our experiments.

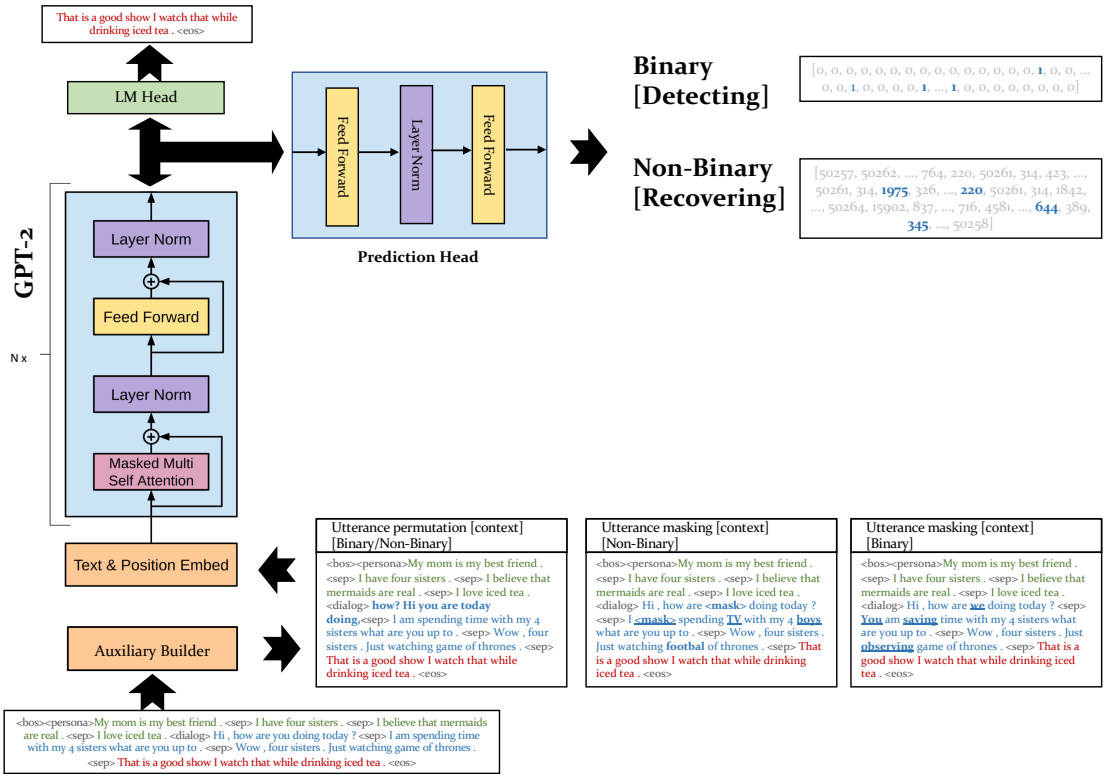


Figure 1: This figure illustrates the auxiliary tasks and the proposed model. The input to the model (our prompt) includes a combination of **persona**, **context**, and the **last conversation**. Each component is separated into special tokens and preceded by a unique token that signifies its component. The model's objective (LM objective) is to generate the final conversational component of the agent's response while disregarding any prior parts.

4.1 Dataset

The experiments in this paper are conducted using two benchmark datasets for open-domain dialogue generation, PersonaChat (Zhang et al., 2018) and DailyDialog (Li et al., 2017). PersonaChat is a large-scale dataset collected by encouraging two individuals to engage in open-domain conversations while exchanging personal information to create personas. The dataset contains over 163,064 utterances (11,907 dialogues) for training and 15,0264 utterances (968 dialogues) for testing. The conversations are naturally diverse, covering various topics and perspectives. In addition, the personal information provided allows the model to generate more informed and coherent responses due to predefined personalities. A true-cased version of PersonaChat[3] is used in the experiments to maintain consistency with the other datasets. On the other hand, the DailyDialog is a small dataset consisting of 13,118 multi-turn dialogues collected from various daily situations. Both datasets are pre-processed to ensure that all conversations are well-formed and coherent and that the data is in a suitable format for training and the auxiliary generator. The datasets are split into training, validation, and test sets for experimentation.

4.2 Baselines

We compared our approach to DialoGPT (Zhang et al., 2020), a Transformer-based response generation model. We design a new prompt that is suitable for our auxiliary tasks. In order to ensure a fair comparison, we fine-tune the GPT-2 model on both of the two multi-turn datasets with the new prompt and using the same configurations introduced by DialoGPT (also known as VanillaGPT-2). This allows us to compare our approach to DialoGPT under the same conditions.

4.3 Implementation Details

Our implementation of both approaches is carried out using PyTorch Lightning[4] and Huggingface Transformers.[5] We train the baseline and our approach on two GPT-2 scales (small 124M and medium 354M parameters). Our approach depends

[3] bavard/personachat_truecased

[4] https://www.pytorchlightning.ai/

[5] https://huggingface.co/

on the dataset we implement the auxiliary tasks on different components persona, context, persona, and context, and by random. All the models are optimized with the AdamW optimizer (Loshchilov and Hutter, 2017) using an initial learning rate of 5e-5 and 3e-5, respectively, for the DailyDialog and PersonaChat datasets, and by using the adaptive learning rate scheduler with 5,000 warm-up steps and weight decay of 0.001. Experiments are performed on NVIDIA A100 for five epochs and a different range of hyperparameters regarding the auxiliary tasks, as seen in Table 1.

Auxiliary Task	α	P_{do}	$P_{reordered}$	P_{masked}	$P_{changed}$
UPD	3.0	0.15	0.1	-	-
UPR	1.0	0.15	0.1	-	-
UMD	3.0	0.15	-	0.8	0.5
UMR	1.0	0.15	-	0.8	0.5

Table 1: Hyperparameters used in the experiments.

4.4 Evaluation Metrics

The assessment of the models is performed in an automated manner utilizing well-established metrics such as perplexity (Vinyals and Le, 2015), BLEU (Papineni et al., 2002), and Rouge-L (Li et al., 2016a). In addition, we also incorporate two additional methods (similarity and correlation with human judgement) for automatic evaluation. These are the Embedding Average (Average), Embedding Extrema (Extrema), and Embedding Greedy (Greedy) metrics (Serban et al., 2017), which provide a deeper understanding of the correspondence between the model's responses and the reference responses. Furthermore, we compute the BertScore (Zhang et al., 2020) and MoverScore (Zhao et al., 2019) to assess the level of agreement between the generated text and human evaluations, and to determine the overall quality of the generated text.

5 Results

Table 2 presents the results of incorporating a combination of objectives and language modeling into various components of open-domain dialog systems. The evaluation was conducted on two benchmark datasets and two different scales of GPT-2. The results demonstrate that the improvement of the model depends on the type of auxiliary objective used in conjunction with language modeling. As demonstrated in the table, utilizing utterance permutation in binary form has a notable impact on reducing the perplexity of the model, with a reduction of 2% being observed. Furthermore, compared to other auxiliary tasks, the use of utterance masking in the binary form leads to improvements in metrics such as BLEU, ROUGE-L, MoverScore, and Extrema. The results also suggest that using auxiliary tasks in larger models improves performance. The comparison between the Persona-Chat dataset highlights the significance of using auxiliary tasks simultaneously in both the Persona and Context components, which results in even better performance. Tables 3 and 4, located in Appendices A and B respectively, present sample generated responses for the two benchmarks, encompassing both the baseline and the optimal auxiliary model.

What is the difference between binary and non-binary auxiliary tasks? The results clearly demonstrate that the model only has access to the left context due to the specific type of attention mechanism employed in GPT (Masked Causal Attention). This limited exposure to context makes it challenging for the model to identify the distorted token correctly. Despite having access to the left context, the model's ability to recognize the scrambled token remains impaired.

What is the impact of implementing these auxiliary tasks on different components of dialogue? Determining the exact part of dialogue that will benefit the most from these tasks is challenging, but it can be agreed upon that combining both the Persona and Context components leads to improved outcomes.

Why do the results vary across these two datasets? The difference can be attributed to the distinct structures of the two benchmarks, as one provides access only to the context and the other to both persona and context.

Does access to both persona and context result in higher quality answers? This depends on the degree to which the persona aligns with the context.

6 Conclusion

In conclusion, our research has focused on improving the quality of generated responses using GPT-2 by proposing two auxiliary tasks. The first task, referred to as utterance permutation, aims to enhance the model's ability to comprehend the interconnections between words in a sentence and

DailyDialog

Scale	Model	PPL	BLEU	ROUGE-L	BERTScore	MoverScore	Average	Greedy	Extrema
SMALL	VanillaGPT-2	11.463	1.188	0.187	0.885	0.045	0.875	0.737	0.881
	UPD [context]	11.445	1.108	0.186	0.884	0.042	0.873	0.735	0.877
	UPR [context]	11.607	0.984	0.179	0.883	0.038	0.870	0.734	0.880
	UMD [context]	11.484	1.365	0.188	0.885	0.047	0.870	0.736	0.881
	UMR [context]	11.859	0.999	0.184	0.884	0.040	0.871	0.735	0.879
MEDIUM	VanillaGPT-2	10.344	2.603	0.208	0.889	0.072	0.880	0.743	0.881
	UPD [context]	9.958	2.393	0.203	0.888	0.064	0.881	0.745	0.883
	UPR [context]	10.192	2.068	0.199	0.887	0.060	0.877	0.739	0.882
	UMD [context]	10.659	2.458	0.208	0.889	0.075	0.879	0.745	0.882
	UMR [context]	10.554	1.886	0.195	0.886	0.055	0.874	0.740	0.880

PERSONA-CHAT

Scale	Model	PPL	BLEU	ROUGE-L	BERTScore	MoverScore	Average	Greedy	Extrema
SMALL	VanillaGPT-2	13.149	1.489	0.099	0.879	0.056	0.878	0.694	0.872
	UPD [persona]	13.100	1.545	0.098	0.878	0.055	0.878	0.694	0.872
	UPD [context]	13.101	1.537	0.098	0.879	0.056	0.878	0.693	0.872
	UPD [persona+context]	13.089	1.426	0.097	0.878	0.054	0.877	0.693	0.872
	UPD [random]	13.108	1.552	0.097	0.878	0.055	0.877	0.693	0.872
	UPR [persona]	13.111	1.489	0.097	0.879	0.055	0.878	0.693	0.872
	UPR [context]	13.132	1.431	0.096	0.878	0.055	0.878	0.693	0.872
	UPR [persona+context]	13.125	1.586	0.097	0.879	0.055	0.878	0.694	0.872
	UPR [random]	13.128	1.427	0.097	0.878	0.054	0.877	0.694	0.872
	UMD [persona]	13.073	1.393	0.098	0.878	0.055	0.878	0.693	0.873
	UMD [context]	13.126	1.538	0.099	0.879	0.056	0.878	0.694	0.873
	UMD [persona+context]	13.079	1.504	0.097	0.878	0.054	0.878	0.692	0.872
	UMD [random]	13.055	1.423	0.096	0.878	0.055	0.877	0.693	0.872
	UMR [persona]	13.309	1.488	0.097	0.878	0.055	0.878	0.693	0.872
	UMR [context]	13.265	1.459	0.098	0.879	0.055	0.878	0.694	0.872
	UMR [persona+context]	13.362	1.371	0.096	0.878	0.053	0.878	0.693	0.872
	UMR [random]	13.263	1.454	0.098	0.878	0.055	0.878	0.694	0.872
MEDIUM	VanillaGPT-2	10.975	1.657	0.100	0.879	0.060	0.878	0.695	0.873
	UPD [persona]	10.969	1.712	0.101	0.880	0.061	0.879	0.696	0.873
	UPD [context]	10.992	1.734	0.101	0.880	0.061	0.879	0.696	0.873
	UPD [persona+context]	10.960	1.693	0.101	0.879	0.060	0.879	0.695	0.873
	UPD [random]	10.978	1.690	0.101	0.879	0.060	0.878	0.694	0.872
	UPR [persona]	10.987	1.703	0.102	0.879	0.060	0.878	0.695	0.873
	UPR [context]	11.006	1.593	0.100	0.879	0.059	0.879	0.695	0.873
	UPR [persona+context]	11.000	1.660	0.100	0.879	0.060	0.879	0.695	0.873
	UPR [random]	11.004	1.575	0.099	0.879	0.059	0.879	0.695	0.873
	UMD [persona]	10.977	1.660	0.101	0.879	0.060	0.879	0.695	0.873
	UMD [context]	11.025	1.659	0.100	0.879	0.060	0.879	0.695	0.873
	UMD [persona+context]	11.004	1.714	0.101	0.879	0.060	0.879	0.696	0.873
	UMD [random]	10.957	1.593	0.099	0.879	0.059	0.879	0.694	0.872
	UMR [persona]	11.063	1.551	0.098	0.879	0.057	0.877	0.693	0.872
	UMR [context]	11.092	1.560	0.099	0.879	0.058	0.878	0.694	0.873
	UMR [persona+context]	11.102	1.468	0.099	0.879	0.057	0.878	0.694	0.873
	UMR [random]	11.044	1.540	0.099	0.879	0.058	0.879	0.695	0.873

Table 2: Results of the evaluation are based on automatic metrics. For each metric, colored numbers indicate the best-performing model.

produce grammatically accurate responses. The second task, utterance masking, is designed to improve the coherence and consistency of the generated responses by challenging the model to predict masked words based on the surrounding context. Our experiments indicate that combining these two auxiliary tasks substantially improves the quality of generated responses. This includes improved grammar, coherence, and consistency, which are crucial aspects of generating high-quality NLP re-

sponses. Furthermore, these results demonstrate the potential of incorporating auxiliary tasks with language models to enhance their performance and improve the quality of generated responses in NLP applications.

Acknowledgements

It is with deep gratitude that we thank the anonymous reviewers for their valuable feedback and guidance, which contributed significantly to the quality and clarity of our work. Finally, this research endeavor would not have been possible without the generous funding of the Wallenberg AI, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation.

We would like to extend our appreciation to the Swedish National Infrastructure for Computing (SNIC), whose latest technologies enabled the computational aspects of our research. Furthermore, we acknowledge the Swedish Research Council's support for SNIC through grant agreement no. 2022/22-1003, which facilitated this research in part.

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's GPT-2 - how can I help you? towards the use of pre-trained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22, Hong Kong. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Guan-Lin Chao and Ian Lane. 2019. BERT-DST: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, Shrimai Prabhumoye, Alan W. Black, Alexander Rudnicky, Jason Williams, Joelle Pineau, Mikhail Burtsev, and Jason Weston. 2020. The second conversational intelligence challenge (ConvAI2). In *The NeurIPS '18 Competition*, pages 187–208, Cham. Springer International Publishing.

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1371–1374, New York, NY, USA. Association for Computing Machinery.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Xiaodong Gu, Kang Min Yoo, and Jung-Woo Ha. 2021. DialogBERT: Discourse-aware response generation via learning to recover and rank utterances. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12911–12919.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*,

pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yi Luan, Chris Brockett, Bill Dolan, Jianfeng Gao, and Michel Galley. 2017. Multi-task learning for speaker-role adaptation in neural conversation models. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 605–614, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Shikib Mehri, Evgeniia Razumovskaia, Tiancheng Zhao, and Maxine Eskenazi. 2019. Pretraining methods for dialog context representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3836–3845, Florence, Italy. Association for Computational Linguistics.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854, Florence, Italy. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Chinnadhurai Sankar, Sandeep Subramanian, Chris Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 32–37, Florence, Italy. Association for Computational Linguistics.

Iulian Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston. 2022. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. In *Deep Learning Workshop at ICML 2015*.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *ArXiv*, abs/1901.08149.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

Yufan Zhao, Can Xu, and Wei Wu. 2020. Learning a simple and effective model for multi-turn response generation with auxiliary tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3472–3483, Online. Association for Computational Linguistics.

A A comparison of Vanilla GPT-2 and GPT-2 with Auxiliary Tasks in Conversational Contexts for the DailyDialogue Dataset

Model	Conversation	Ground response	Generated response
VanillaGPT-2 [sm]	- By the way miss, where is the toilet? - Toilets are in the rear, I am afraid all the toilets are fully occupied at the moment. - What? Oh, what we live! Thank you very much for your help, miss.	You are welcome.	- Thanks. - I'll see you in Beijing in a few days. - Goodbye, bye.
UPD [sm][context]	- By the way miss, where is the toilet? - Toilets are in the rear, I am afraid all the toilets are fully occupied at the moment. - What? Oh, what we live! Thank you very much for your help, miss.	You are welcome.	- Don't mention it. - You are welcome. - You are welcome.
VanillaGPT-2 [md]	- Yeah, look at all those new buildings going up! Mr. Zhang, the traffic is pretty smooth. But we were told the roads from the airport to downtown were quite crowded and traffic jams could be as long as half an hour. - Yes, they were. But it has already past. The traffic from the airport to downtown has been relieved after the completion of Yan'an Aerial Road. - We've never expected that Shanghai is such a big metropolis, same as New York city. David, would it be a good idea to rent a car here? I've got an international driver's license with a pretty good record in the States.	That's wonderful. OK. I'll remember that. But I have to warn you that you must drive very carefully in the downtown area, for there're no separate lanes for motor vehicles and bicycles in some places, and still quite a few people don't take road regulations seriously.	- Oh, yeah, you'd better have it. - Oh, no, that's stupid. You'd better drive a taxi! - That's a good idea.
UPD [md][context]	- Yeah, look at all those new buildings going up! Mr. Zhang, the traffic is pretty smooth. But we were told the roads from the airport to downtown were quite crowded and traffic jams could be as long as half an hour. - Yes, they were. But it has already past. The traffic from the airport to downtown has been relieved after the completion of Yan'an Aerial Road. - We've never expected that Shanghai is such a big metropolis, same as New York city. David, would it be a good idea to rent a car here? I've got an international driver's license with a pretty good record in the States.	That's wonderful. OK. I'll remember that. But I have to warn you that you must drive very carefully in the downtown area, for there're no separate lanes for motor vehicles and bicycles in some places, and still quite a few people don't take road regulations seriously.	- Mr. Zhang, I can assure you that the traffic isn't that bad here. There's a traffic island at the edge of the city to ease the traffic. But I still prefer to drive my own car. - Mr. Zhang, I can assure you that the traffic isn't that bad here. There's a traffic island at the edge of the city to ease the traffic. But I still prefer to drive my own car. - That would be great!

Table 3: A comparison of Vanilla GPT-2 and Utterance Permutations (binary) models with small and medium scales, demonstrating ground truth responses and generated responses for various dialogues. Overall, the responses are more or less related to the context, but the GPT-2 with the auxiliary tasks produces responses that are more closely related to the context.

B A comparison of Vanilla GPT-2 and GPT-2 with Auxiliary Tasks in Conversational Contexts for the PersonaChat Dataset

Model	Persona	Conversation	Ground response	Generated responses
VanillaGPT-2 [sm]	- I love to drink wine and dance in the moonlight. - I am very strong for my age. - I'm 100 years old. - I feel like I might live forever.	- Hi, how are you doing today? - Good! you! Celebrating with fellow centenarians. - Nice. I'm playing some card games with my family. - That sounds like fun. I like wine and dancing too! - Same. I try to get a small workout in a three mile walk for me is key. - Awesome! Walking like that you might live forever like me!! - That's true. My favorite way to unwind is reading a book.	You sound very like minded, just add some wine.	- I know! Where do you live? I'm still 100 years from retirement! - What is your favorite book? Do you like to read? - That's a big idea! I love books too.
VanillaGPT-2 [sm]	- I'm now looking for a new job. - My favorite movie is Friday. - I am a doctor. - My daughter is a child prodigy.	- Hello, I'm sitting here with my dog. How are you?	I'm well friend. Looking for new employment at the moment.	- Hi my name is Dr. Robert. - Hi how are you today. - Hi. how are you doing?
UMB [sm][random]	- I love to drink wine and dance in the moonlight. - I am very strong for my age. - I'm 100 years old. - I feel like I might live forever.	- Hi, how are you doing today? - Good! you! Celebrating with fellow centenarians. - Nice. I'm playing some card games with my family. - That sounds like fun. I like wine and dancing too! - Same. I try to get a small workout in a three mile walk for me is key. - Awesome! Walking like that you might live forever like me!! - That's true. My favorite way to unwind is reading a book.	You sound very like minded, just add some wine.	- That sounds like a lot of fun. Very healthy. I'm really strong. You should try to eat healthy! Books are fun. I have tons.
UMB [sm][random]	- I'm now looking for a new job. - My favorite movie is Friday. - I am a doctor. - My daughter is a child prodigy.	- Hello, I'm sitting here with my dog. How are you?	I'm well friend. Looking for new employment at the moment.	- I'm doing good. Just finished my shift at the hospital. How are you? - I'm doing well, and yourself? - I am well, how about yourself.
VanillaGPT-2 [md]	- I prefer being inside. - I am a recording engineer and own my own studio. - Whats up I am a 46 year old single dad 46 a kids. - I live in California but the recording artist market was dead last year.	- Hello, call of duty is my favorite game. Do you game?	Yes I do after all I am a engineer.	- Not as much as my daughter she's my baby. - No I am just a big time recording engineer. - I prefer the movie games and tv shows. I have a studio in California.
UMB [md][random]	- I prefer being inside. - I am a recording engineer and own my own studio. - Whats up I am a 46 year old single dad 46 a kids. - I live in California but the recording artist market was dead last year.	- Hello, call of duty is my favorite game. Do you game?	Yes I do after all I am a engineer.	- I don't play game like that. I'm a live video coordinator. - Not really. What do you do? - I am not sure I've the computer but the tv show is my favorite.

Table 4: An examination of Vanilla GPT-2 and Binary Utterance Masking (random in context and persona) models in small and medium sizes, showcasing authentic responses. The responses generally have a degree of contextual and persona relevance, but when the GPT-2 is integrated with auxiliary tasks, the responses demonstrate a stronger connection to the context and persona.

Uncertainty-Aware Natural Language Inference with Stochastic Weight Averaging

Aarne Talman* **Hande Celikkanat*** **Sami Virpioja***
Markus Heinonen† **Jörg Tiedemann***

* Department of Digital Humanities, University of Helsinki
name.surname@helsinki.fi
† Department of Computer Science, Aalto University
markus.o.heinonen@aalto.fi

Abstract

This paper introduces Bayesian uncertainty modeling using Stochastic Weight Averaging-Gaussian (SWAG) in Natural Language Understanding (NLU) tasks. We apply the approach to standard tasks in natural language inference (NLI) and demonstrate the effectiveness of the method in terms of prediction accuracy and correlation with human annotation disagreements. We argue that the uncertainty representations in SWAG better reflect subjective interpretation and the natural variation that is also present in human language understanding. The results reveal the importance of uncertainty modeling, an often neglected aspect of neural language modeling, in NLU tasks.

1 Introduction

Arguably, human language understanding is not objective nor deterministic. The same utterance or text can be interpreted in different ways by different people depending on their language standards, background knowledge and world views, the linguistic context, as well as the situation in which the utterance or text appears. This uncertainty about potential readings is typically not modeled in Natural Language Understanding (NLU) research and is often ignored in NLU benchmarks and datasets. Instead, they usually assign a single interpretation as a gold standard to be predicted by an artificial system ignoring the inherent ambiguity of language and potential disagreements that humans arrive at.

Some datasets like SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) do, however, contain information about different readings in the form of annotation disagreement. These datasets include the labels from five different rounds of annotation which show in some cases clear disagreement about the correct label for the sentence pair. Those labeling discrepancies can certainly be a result of annotation mistakes but more commonly they arise from differences in understanding the task, the given information and how it relates to world knowledge and personal experience.

Moving towards uncertainty-aware neural language models, we present our initial results using Stochastic Weight Averaging (SWA) (Izmailov et al., 2018) and SWA-Gaussian (SWAG) (Maddox et al., 2019) on the task of Natural Language Inference. SWAG provides a scalable approach to calibrate neural networks and to model uncertainty presentations and is straightforward to apply with standard neural architectures. Our study addresses the two main questions:

- How does uncertainty modeling using SWAG influence prediction performance and generalization in NLI tasks?
- How well does the calibrated model reflect human disagreement and annotation variance?

In this paper, we first test the performance of SWA and SWAG in SNLI and MNLI tasks. We then study if adding weight averaging improves the generalization power of NLI models as tested through cross-dataset experiments. Finally, we analyse the probability distributions from SWA and SWAG to test how well the model uncertainty corresponds to annotator disagreements.

2 Background and Related Work

2.1 Uncertainty in human annotations

In a recent position paper Plank (2022) argue that instead of taking human label variation as a problem, we should embrace it as an opportunity and

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 358–365

take it into consideration in all the steps of the ML pipeline: data, modeling and evaluation. The paper provides a comprehensive survey of research on (i) reasons for human label variation, (ii) modeling human label variation, and (iii) evaluating with human label variation.

Pavlick and Kwiatkowski (2019) studied human disagreements in NLI tasks and argue that we should move to an evaluation objective that more closely corresponds to the natural interpretation variance that exists in data. Such a move would require that NLU models be properly calibrated to reflect the distribution we can expect and, hence, move to a more natural inference engine.

Chen et al. (2020) propose Uncertain NLI (UNLI), a task that moves away from categorical labels into probabilistic values. They use a scalar regression model and show that the model predictions correlate with human judgement.

2.2 Representing Model Uncertainty

The approach to uncertainty modeling that we consider is related to the well-established technique of model ensembling. Stochastic optimization procedures applied in training deep neural networks are non-deterministic and depend on hyper-parameters and initial seeds. Ensembles have been used as a pragmatic solution to average over several solutions, and the positive impact on model performance pushed ensembling into the standard toolbox of deep learning. Related to ensembling is the technique of checkpoint averaging (refer to e.g. Gao et al., 2022), which is also known to improve performance.

Intuitively, ensembles and checkpoint averages also reflect the idea of different views and interpretations of the data and, therefore, provide a framework for uncertainty modeling. Stochastic Weight Averaging (SWA, Izmailov et al. (2018)) and SWA-Gaussian (SWAG, Maddox et al. (2019)) both build on this idea. SWA proposes using the first moments of the parameters of the solutions traversed by the optimizer during the optimization process, as mean estimates of the model parameters. Using such mean values have been argued to result in finding wider optima, providing better generalization to unseen data. On top of these mean estimations procured by SWA, SWAG then adds a low-rank plus diagonal approximation of covariances, which, when combined with the aforementioned mean estimations, provide us with corresponding Gaussian posterior approximations over model parameters. Posterior distribution approximations learned this way then represent our epistemic uncertainty about the model (Kiureghian and Ditlevsen, 2009), meaning the uncertainty stemming from not knowing the perfect values of the model parameters, since we do not have infinite data to train on. During test time, instead of making estimates from a single model with deterministic parameters, we sample N different models from the approximated posteriors for each model parameter, and use the average of their prediction distributions as the model response.

Note that as both of these methods use the optimizer trajectory for the respective approximations, they provide significant computational efficiency as compared to the vanilla ensembling baseline. In this paper, we use SWA mainly as another baseline for SWAG, which needs to outperform SWA in order to justify the additional computation required for the covariance approximation.

SWA (Izmailov et al., 2018) is a checkpoint averaging method that tracks the optimization trajectory for a model during training, using the average of encountered values as the eventual parameters:

$$\theta_{\text{SWA}} = \frac{1}{T}\sum_{i=1}^{T}\theta_i, \tag{1}$$

with θ_{SWA} denoting the SWA solution for parameter θ after T steps of training.[1]

SWAG (Maddox et al., 2019) extends this method to estimate Gaussian posteriors for model parameters, by also estimating a covariance matrix for the parameters, using a low-rank plus diagonal posterior approximation. The diagonal part is obtained by keeping a running average of the second uncentered moment of each parameter, and then at the end of the training calculating:

$$\Sigma_{\text{diag}} = \text{diag}(\frac{1}{T}\sum_{i=1}^{T}\theta_i^2 - \theta_{\text{SWA}}^2) \tag{2}$$

while the diagonal part is approximated by keeping a matrix DD^{T} with columns $D_i = (\theta_i - \hat{\theta}_i)$, $\hat{\theta}_i$ standing for the running estimate of the parameters' mean obtained from the first i samples. The rank of the approximation is restricted by keeping only the final K-many of the D_i vectors, and dropping the previous, with K being a hyperparameter

[1] In this work, we use one sample from each epoch.

of the method:

$$\Sigma_{\text{low-rank}} \approx \frac{1}{K-1} DD^{\intercal} \quad (3)$$

$$= \frac{1}{K-1} \sum_{i=T-K+1}^{T} (\theta_i - \hat{\theta}_i)(\theta_i - \hat{\theta}_i)^{\intercal} \quad (4)$$

The overall posterior approximation is given by:

$$\theta_{\text{SWAG}} \sim \mathcal{N}(\theta_{\text{SWA}}, \frac{1}{2}(\Sigma_{\text{diag}} + \Sigma_{\text{low-rank}})). \quad (5)$$

Once the posteriors are thus approximated, in test time, the model is utilized by sampling from the approximated posteriors for N times, and taking the average of the predicted distributions from these samples as the answer of the model.

One of the advantages of SWAG is the possibility to seamlessly start with any pre-trained solution. Approximating the posterior is then done during fine-tuning without the need to change the underlying model.

2.3 Stochastic Weight Averaging in NLP

Previous work on Stochastic Weight Averaging in the context of NLP is very limited. Lu et al. (2022) adapt SWA for pre-trained language models and show that it works on par with state-of-the-art knowledge distillation methods. Khurana et al. (2021) study pre-trained language model robustness on a sentiment analysis task using SWA and conclude that SWA provides improved robustness to small changes in the training pipeline. Kaddour et al. (2022) test SWA on multiple GLUE benchmark tasks (Wang et al., 2018) and find that the method does not provide clear improvement over the baseline.

Maddox et al. (2019) test SWAG in language modeling tasks using Penn Treebank and WikiText-2 datasets and show that SWAG improves test perplexities over a SWA baseline. To the best of our knowledge our work is the first to apply SWAG to NLU tasks.

3 Experiments

We test the performance of SWA and SWAG on the natural language inference task using three NLI datasets, including cross-dataset experiments, and study the effect on both hard and soft labeling. Code for replicating the experiments is available on GitHub: `https://github.com/Helsinki-NLP/uncertainty-aware-nli`

3.1 Datasets

We use Stanford Natural Language Inference corpus (SNLI) (Bowman et al., 2015) and Multi-Genre Natural Language Inference (MNLI) corpus (Williams et al., 2018) as the datasets in our experiments. We also study cross-dataset generalisation capability of the model with and without weight averaging. For those experiments we also include SICK (Marelli et al., 2014) as a test set. In cross-dataset generalization experiments we first fine-tune the model with a training data from one NLI dataset (e.g. SNLI) and then test with a test set from another NLI dataset (e.g. MNLI-mm).

SNLI The Stanford Natural Language Inference (SNLI) corpus is a dataset of 570k sentence pairs which have been manually labeled with entailment, contradiction, and neutral labels. The source for the premise sentences in SNLI were image captions from the Flickr30k corpus (Young et al., 2014).

MNLI The Multi-Genre Natural Language Inference (MNLI) corpus is made of 433k sentence pairs labeled with entailment, contradiction and neutral, containing examples from ten genres of written and spoken English. Five of the genres are included in the training set. The development and test sets have been split into matched (MNLI-m) and mismatched (MNLI-mm) sets, where the former includes only sentences from the same genres as the training data, and the latter includes genres not present in the training data.[2] The MNLI dataset was annotated using very similar instructions as for the SNLI dataset and, therefore it is safe to assume that the definitions of entailment, contradiction and neutral are the same across these two datasets.

SICK SICK is a dataset that was originally designed to test compositional distributional semantics models. The dataset includes 9,840 examples with logical inference (negation, conjunction, disjunction, apposition, relative clauses, etc.). The

[2] As the test data for MNLI have not been made publicly available, we use the development sets when reporting the results for MNLI.

dataset was constructed automatically by taking pairs of sentences from a random subset of the 8K ImageFlickr (Young et al., 2014) and the SemEval 2012 STS MSRVideo Description (Agirre et al., 2012) datasets by using rule-based approach to construct examples for the different logical inference types.

3.2 Methods

In all the experiments we fine tune a pre-trained RoBERTa-base model (Liu et al., 2019) from the Hugging Face Transformers library (Wolf et al., 2020). As a common practice in the NLI tasks, we use the majority-vote gold labels for training.

We add stochastic weight averaging to the RoBERTa model by using the SWA implementation from PyTorch 1.12[3] and the SWAG implementation by (Maddox et al., 2019)[4]. To study how well SWA and SWAG perform in NLI as compared to a baseline model, we ran the same fine-tuning with SNLI and MNLI datasets, while utilizing SWA and SWAG for mean and variance estimations of parameters undergoing fine-tuning.

3.3 Results

The standard evaluation for the NLI task is the accuracy on aggregated gold labels. However, as two of the test data sets (from SNLI and MNLI) also contains multiple human annotations, we also use those for measuring the cross entropy of the predicted distribution on the human label distribution (soft labeling, e.g. Peterson et al., 2019; Pavlick and Kwiatkowski, 2019).

3.3.1 Accuracy

The basic classification results are in Table 1. We report average accuracies and standard deviation over 5 runs with different random seeds.

Both SWA and SWAG provide clear improvements over the baseline without weight averaging. SWAG performs slightly better than SWA across all the three experiments.

In order to test if weight averaging improves the generalization capability of NLI models, we further performed cross-dataset generalization tests following (Talman and Chatzikyriakidis, 2019). The results are reported in Table 2.

Dataset	Method	Acc (%)	SD	Δ
SNLI	base	90.80	0.26	-
SNLI	SWA	91.47	0.24	+0.67
SNLI	SWAG	**91.59**	0.14	**+0.79**
MNLI-m	base	86.53	0.20	-
MNLI-m	SWA	87.60	0.19	+1.07
MNLI-m	SWAG	**87.76**	0.12	**+1.23**
MNLI-mm	base	86.31	0.26	-
MNLI-mm	SWA	87.34	0.29	+1.03
MNLI-mm	SWAG	**87.51**	0.19	**+1.20**

Table 1: Comparison of SWA and SWAG performance on NLI benchmarks (mean accuracy and standard deviation over 5 runs). Δ is the difference to the baseline result (base) with no weight averaging.

Dataset	Method	Acc (%)	SD	Δ
SNLI → MNLI-m	base	77.31	0.57	
SNLI → MNLI-m	SWA	**79.67**	0.37	**2.36**
SNLI → MNLI-m	SWAG	79.33	0.21	2.02
SNLI → MNLI-mm	base	77.40	0.78	
SNLI → MNLI-mm	SWA	**79.44**	0.19	**2.04**
SNLI → MNLI-mm	SWAG	79.24	0.29	1.84
SNLI → SICK	base	57.08	0.77	
SNLI → SICK	SWA	57.09	0.32	0.01
SNLI → SICK	SWAG	**57.17**	0.37	**0.08**
MNLI → SNLI	base	82.84	0.74	
MNLI → SNLI	SWA	84.15	0.35	1.31
MNLI → SNLI	SWAG	**84.45**	0.27	**1.61**
MNLI → SICK	base	**56.63**	0.94	
MNLI → SICK	SWA	56.17	0.60	-0.46
MNLI → SICK	SWAG	56.53	0.91	-0.10

Table 2: Cross-dataset experiments with and without weight averaging (mean accuracy and standard deviation over 5 runs with different random seeds), where the left hand side of the arrow is the training set and the right hand side is the testing set.

The results of cross-dataset experiments are slightly mixed: We do not notice a clear advantage of SWAG over SWA, but with the exception of training with MNLI and testing with SICK, we do notice improvement for weight averaging approaches as compared to the baseline. The performance on SICK drops significantly in all cases and the difference between the approaches is minimal, showing that the NLI training data is not a good fit for that benchmark. The other cross-dataset results highlight the advantage of stochastic weight averaging, which is in line with the findings of (Izmailov et al., 2018) that the method is able to locate wider optima regions with better generalization capabilities.

[3] https://pytorch.org/docs/1.12/optim.html#stochastic-weight-averaging

[4] https://github.com/wjmaddox/swa_gaussian

Dataset	Method	Cross Entropy	Δ
SNLI	base	0.83	
SNLI	SWA	0.75	-0.08
SNLI	SWAG	**0.69**	**-0.14**
MNLI-m	base	0.87	
MNLI-m	SWA	0.80	-0.07
MNLI-m	SWAG	**0.73**	**-0.14**
MNLI-mm	base	0.84	
MNLI-mm	SWA	0.77	-0.07
MNLI-mm	SWAG	**0.69**	**-0.15**
SNLI → MNLI-m	base	1.13	
SNLI → MNLI-m	SWA	0.90	-0.23
SNLI → MNLI-m	SWAG	**0.80**	**-0.33**
SNLI → MNLI-mm	base	1.12	
SNLI → MNLI-mm	SWA	0.88	-0.24
SNLI → MNLI-mm	SWAG	**0.79**	**-0.33**
MNLI → SNLI	base	1.04	
MNLI → SNLI	SWA	0.97	-0.07
MNLI → SNLI	SWAG	**0.89**	**-0.15**

Table 3: Comparison of cross entropies between data annotation distributions using base, SWA and SWAG methods. Δ is the difference to the baseline cross entropy values.

3.3.2 Cross Entropy

We also test how well these weight averaging and covariance estimating methods help towards better modeling annotator disagreement and annotation uncertainty in the NLI testsets of SNLI and MNLI. These two datasets come with five different annotation labels for every data point, often with high disagreement between human annotators, indicating inherently *confusing* data points with high aleatoric uncertainty (Kiureghian and Ditlevsen, 2009). For quantifying the goodness of fit of the model predictions, we calculate the cross entropy between the predicted and annotation distributions.[5]

Table 3 depicts the resulting cross entropy values, with lower values denoting more faithful predictions. SWA and SWAG result in consistently more similar distributions with that of annotations, complementing their overall better accuracy results (Section 3.3). In contrast to the accuracy results, here SWAG outperforms SWA in all cases, indicating that the modeling uncertainty through the approximation of Gaussian posteriors helps to model annotator disagreements more accurately. The results also carry over to the cross-dataset experiments as shown on the table.

The comparison between system predictions and annotator variation deserves some further analysis. Preliminary study (refer to examples in Appendix A) indicates that the prediction uncertainty in SWAG for individual instances very well follows human annotation confusion. Furthermore, we identified cases with a larger mismatch between system predictions and human disagreement where the latter is mainly caused by erroneous or at least questionable decisions. This points to the use of SWAG in an active learning scenario, where annotation noise can be identified using a well calibrated prediction model.

[5]Note that for the Baseline and SWA models, we consider the output from the eventual softmax function as the predicted distribution, while for the SWAG model, we use the average output distribution from $N = 20$ sampled models.

4 Conclusions

Our results show that weight averaging provides consistent and clear improvement for both SNLI and MNLI datasets. The cross-dataset results are slightly mixed but also show the trend of improved cross-domain generalization. Finally, we demonstrate a clear increase in the correlation with human annotation variance when comparing SWAG with non-Bayesian approaches.

For future work we consider making use of multiple annotations also during training and extensions of SWAG such as MultiSWAG (Wilson and Izmailov, 2020). We also plan to test the methods on different NLU datasets, especially those with a high number of annotations (e.g. Nie et al., 2020), and compare the annotation variation and system predictions in more detail. Finally, in our future work we will explore other uncertainty modeling techniques, like MC dropout (Gal and Ghahramani, 2016), in NLU and see how they compare with stochastic weight averaging techniques.

Acknowledgements

This work is supported by the ICT 2023 project "Uncertainty-aware neural language models" funded by the Academy of Finland (grant agreement 345999) and the FoTran project, funded by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement no. 771113). We also wish to acknowledge CSC – The Finnish IT Center for Science for the generous computing resources they have provided.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tongfei Chen, Zhengping Jiang, Adam Poliak, Keisuke Sakaguchi, and Benjamin Van Durme. 2020. Uncertain natural language inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8772–8779, Online. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Yingbo Gao, Christian Herold, Zijian Yang, and Hermann Ney. 2022. Revisiting checkpoint averaging for neural machine translation. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 188–196, Online only. Association for Computational Linguistics.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pages 876–885. Association For Uncertainty in Artificial Intelligence (AUAI).

Jean Kaddour, Linqing Liu, Ricardo Silva, and Matt Kusner. 2022. When do flat minima optimizers work? In *Advances in Neural Information Processing Systems*.

Urja Khurana, Eric Nalisnick, and Antske Fokkens. 2021. How emotionally stable is ALBERT? testing robustness with stochastic weight averaging on a sentiment analysis task. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 16–31, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Peng Lu, Ivan Kobyzev, Mehdi Rezagholizadeh, Ahmad Rashid, Ali Ghodsi, and Phillippe Langlais. 2022. Improving generalization of pre-trained language models via stochastic weight averaging. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4948–4954, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. 2019. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Joshua C. Peterson, Ruairidh M. Battleday, Thomas L. Griffiths, and Olga Russakovsky. 2019. Human uncertainty makes classification more robust. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9616–9625.

Barbara Plank. 2022. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aarne Talman and Stergios Chatzikyriakidis. 2019. Testing the generalization power of neural network models across NLI benchmarks. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages

85–94, Florence, Italy. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Andrew G Wilson and Pavel Izmailov. 2020. Bayesian deep learning and a probabilistic perspective of generalization. In *Advances in Neural Information Processing Systems*, volume 33, pages 4697–4708. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

A Appendix

Here we showcase and discuss three randomly selected data points from the SNLI dataset, and compare the predictions of the $N = 20$ samples from the SWAG model with the annotation distributions for each of these points. Table 4 presents two cases (left and middle) in which the SWAG model makes the correct prediction, and another case (right) in which the model makes an incorrect prediction. In the high agreement cases, indicated by lower cross entropies between the annotations and prediction, the SWAG model not only selects the correct label for the instance, but also predicts the annotator disagreement correctly when such a disagreement exists (middle) versus when it does not (left).

The third figure presents a case where the predictions of the SWAG samples are more *certain* than expected: Annotators disagree on whether the hypothesis is entailment or neutral, whereas the model predictions place all probability mass to the neutral class. The corresponding cross entropy is high, which reflects this disagreement. It should be noted that this is also a fairly controversial and difficult data point, and to conclude Entailment requires making some strong assumptions. Ideally, such disagreements between system predictions and annotator distributions may also be used as cues within the training process itself. Two potential venues are (1) using the incongruence between the two distributions as the loss signal to drive the optimization process directly (as opposed to using only the gold label and the predicted class label), and (2) using the incongruence in predictions in an active learning scenario.

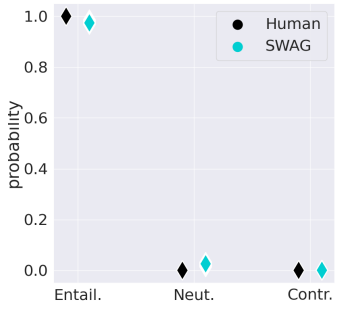
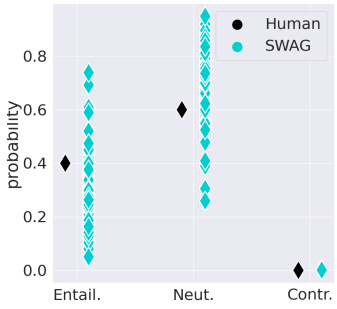
	Correct Prediction		Incorrect Prediction
			
Premise	Two female martial artists demonstrate a kick for an audience.	A group of boys playing street soccer.	People shopping at an outside market.
Hypothesis	Two martial artists demonstrate moves for the audience.	A team is playing street soccer.	People are enjoying the sunny day at the market.
Annotations	E-E-E-E-E	E-E-N-N-N	E-E-E-N-N
Cross-Ent	0.02	0.69	3.88

Table 4: Comparison of probability distributions of human annotations vs. SWAG model predictions, for three randomly selected data points from the SNLI dataset. *(Left and middle)* Correctly predicted cases, as indicated by low cross entropy, *(Right)* A incorrectly predicted case, as indicated by high cross entropy. SWAG points indicate the outputted probability distributions from $N = 20$ samples.

Alignment of Wikidata lexemes and Det Centrale Ordregister

Finn Årup Nielsen
DTU Compute
Technical University of Denmark
faan@dtu.dk

Abstract

Two Danish open access lexicographic resources have appeared in recent years: lexemes in Wikidata and Det Centrale Ordregister (COR). The lexeme part of Wikidata describes words in different languages and COR associates an identifier with each different form of Danish lexemes. Here I described the current state of the linking Wikidata lexemes with COR and some of the problems encountered.

1 Wikidata

Wikidata (Vrandečić and Krötzsch, 2014), the large collaboratively edited knowledge graph, has a special namespace for words and the first lexeme in this namespace was created in 2018 (Nielsen, 2019).[1] Since then the number of lexemes has steadily grown. In February 2020, Wikidata had over 250,000 lexemes (Nielsen, 2020). As of January 2023, there were over 980,000 lexemes, over 12 million forms and over 290,000 senses. Wikidata records, e.g., over 213,000 German lexemes and over 16,000 Danish lexemes. A lexeme in Wikidata describes forms and senses with a growing set of properties. Each lexeme, form, and sense has a unique identifier. An example of a property for Danish lexemes is the link to words in the Danish wordnet resource DanNet (Pedersen et al., 2009).

Four templates for Lucas Werkmeister's *Wikidata Lexeme Forms* tool[2] are so far set up to help enter Danish words in Wikidata: For adjectives, verbs, and common and neuter gender nouns.

[1]The Sumerian word for mother, https://www.wikidata.org/w/index.php?title=Lexeme:L1&action=history&dir=prev

[2]https://lexeme-forms.toolforge.org

2 Det Centrale Ordregister

Det Centrale Ordregister (COR) (Dideriksen et al., 2022) is a Danish lexicographic resource with an initial test dataset released in May 2022 and since then updated. The resource and its description are available at https://ordregister.dk/. COR mostly describes the forms of lexemes and gives each form an identifier. Work on the semantic part of COR is underway (Nimb et al., 2022; Pedersen et al., 2022), but here I will not consider this part. The version of COR I consider is version 1.02 of the core COR and the 1.0 version for COR-EXT.

3 COR Wikidata properties

For linking Wikidata and COR, the Wikidata community has created two Wikidata properties for COR identifiers: One for the lexeme and one for the form. For instance, for the Wikidata lexeme *Sudan* (L993787) the COR lemma ID, level 1 identifier (P10831) is "COR.09978" while the COR form, level 1 identifiers (P10830) are "COR.09978.500.01" and "COR.09978.500.01" for the non-genitive (*Sudan*) and the genitive (*Sudans*) forms, respectively. The current regular expression contraint for the lexeme form is `COR\.(EXT\.\d{6}|\d{5})` allowing the core COR as well as the level 2 COR-EXT identifiers, — and it could be extended if other resources appear. The two COR Wikidata properties were created in June 2022. As of February over 2,100 COR lexemes/lemmas and over 1,800 COR forms are linked from Wikidata.[3] Around 390 Danish Wikidata lexemes have so far been annotated as not being present in COR. Most of these lexemes are compounds. Proper nouns and a few interjections comprise most of the rest.

[3]https://ordia.toolforge.org/statistics/

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 366–370

4 What is a lexeme and a form?

For linking Wikidata and COR it is important that there is a correspondence between the items of the two resources.

Danish words may have spelling variants, e.g., *højtaler* and *højttaler*. In COR, they are grouped under the same lexeme and as different variants. In Wikidata, spelling variants may be grouped under the same form. For instance, the English form L1347-F1 is currently listing both *color* and *colour*, separating them with different language specifications (*en* and *en-gb*). In Danish, the spelling variants do not arise due to different languages and the current Wikidata interface cannot handle spelling variants within the same language in one form. So far the Danish lexemes in Wikidata create separate forms for each spelling variants, e.g., L229388-F1 is *højtaler* and L229388-F2 is *højttaler*, making one COR spelling variant map to one Wikidata form. Two Wikidata spelling variant forms can be linked with the symmetric *alternative form* property (P8530).

In Wikidata, we have so far followed the scheme of *Den Danske Ordbog* (DDO) for Danish nouns with multiple genders and use only one lexeme for these cases. For instance, *øl* has common and neuter versions of the noun under the same lemma in DDO.[4] In Wikidata, this is also one lexeme: L39743. In COR there are two lexemes for *øl*: COR.45830 and COR.48125, thus in this case we get a one-to-two relationship between Wikidata and COR. Other examples of this type are *vand* and *kirsebær*. COR has *safran* (COR.93857) also as both common and neuter gender represented with one lexeme. The indefinite form, which does not reveal the gender, has two forms in COR: COR.93857.110.01 and COR.93857.120.01. Such a word often occurs in the bare form with no article or morphological gender suffix, so it may be impossible to detect the gender of the form in the context. Lexeme linking will have an ambiguity in this case. The current entry in Wikidata has just one (non-genitive) indefinite form. The same issue appears, e.g., for *kanel* (COR.57435) and *druk* (COR.86399).

Homographs that only have one gender are well-aligned between COR, Wikidata, and DDO, e.g., the noun(s) *fyr* has 3 separate lexemes in Wikidata and also has 3 separate lexemes (lemmas) in COR.

Superlative may be regarded as a derivation from the positive form (Hansen and Heltoft, 2019, p. 186–7), but both in COR and the Danish lexemes in Wikidata, the superlative is forms under normal adjective lexemes that also has the positive and comparative forms.

Centaur nouns (Danish: kentaurnominaler) are developed from verbs with an *-en* suffix. In both Wikidata and in COR (i.e., COR-EXT), they are regarded as separate lexemes, — and not a form of the verb lexeme.

Danish *perfektum participium* (adjective forms of the verb) exists in the borderland between being forms of a verb and an adjective derived from a verb (Holm and Christensen, 2019, p. 118). So should *perfektum participium* forms be grouped under a separate lexeme? COR does usually not record the adjective form of the verb separate from the verb. For instance, *barberet* in *et barberet ansigt* (a shaved face, COR.38323.213.01) is grouped under the verb *barbere* (COR.38323). There is an advantage in making a derived lexeme for the adjective forms, as it allows for the description of the sense, e.g., for *barberet* the antonym *ubarberet* can be specified. In Wikidata, the sense of the adjective *barberet* (L940943) is linked via the antonym property (P5974) to the sense of *ubarberet* (L940942) and vice versa. If the adjective was not a separate lexeme, but just part of the verb lexeme, it would not be straightforward to make this antonym link. With the adjective *barberet* as an individual lexeme, words such as *glatbarberet* and *nybarberet* becomes compounds, and the senses of the two compounds can be linked to the sense of *barberet* via the hypernym property.

In COR, it is not all verb-derived *-et* adjectives that are not separated from a corresponding verb, e.g., *snobbet* (COR.24113) is separate from the verb *snobbe* (COR.37973). In these cases, the verb still has the *perfektum participium* forms. Surprisingly, *overstimulere* is in COR as a verb (COR.32555) and *overstimuleret* is not in COR as a separate adjective, while *understimulere* is not in COR, while *understimuleret* is a separate adjective (COR.23107).

Adverbs from adjectives are grouped under the adjective lexeme in COR. In Wikidata, lexemes must be associated with a single lexical category (e.g., adverb or adjective). We have created separate lexemes for Danish adverbs from adjectives

[4] https://ordnet.dk/ddo/ordbog?query=%C3%B8l.

in Wikidata (for the few entered so far), e.g., the adverb *hurtigt* with forms *hurtig*, *hurtigere* and *hurtigst* has the Wikidata lexeme identifier (L691405) that is separate from the adjective lexeme (L42201) and with the COR lemma identifier (COR.15444) duplicated between the two Wikidata items.

5 Lexical categories

Below I will go through some of the most important lexical categories and how Wikidata and COR aligns. Small lexical categories such as conjunctions and prepositions are fully linked, with a few oddities: DDO and Wikidata have *dels* as a conjunction. It is an adverb in COR. Wikidata has *plus at* as a conjunction. It does not exist in COR.

5.1 Pronouns

What is a form and a lexeme for a pronoun varies between resources. For instance, COR has *han*, *ham* and *hans* (he, him, his) collected in one lexeme (COR.01880), while DDO separates them among three different dictionary entries. In Wikidata we have followed DDO and have three different lexemes for the three Danish words. In the English part of Wikidata, *he*, *him*, *his* and *himself* are collected into one lexeme (L485).

There are unusual grammatical features for some of the forms in COR, e.g., *ingens* has two forms indicated with "pron.gen" (pronoun, genitive) and "pron.sg.fk.gen" (pronoun, singular, common gender, genitive). Currently, the former links to the plural genitive form in Wikidata.

5.2 Numeral

COR has genitive forms for numerals, while the few numerals in COR-EXT do not. We started with no genitive forms for Danish numerals in Wikidata but now have begun to enter them. A few numerals in COR-EXT includes Arabic numerals, e.g., "94" as a form (COR.EXT.129099.600.02). Arabic numerals do not appear in the core COR and we have so far not included them in Wikidata.

5.3 Nouns

The non-genitive forms are not annotated as explicitly non-genitive in COR. In Wikidata, we can explicitly annotate forms that are not genitive with the non-genitive item (Q98946930).

Some forms are recorded in CORs but should perhaps not be present, while other forms are not recorded but perhaps should be. For instance, *drukning* (COR.61517) has no plural forms in COR while Den Danske Ordbog (DDO)[5] records plural forms and an Internet search[6] returns some examples of the plural form. *forundring* has neither plural forms in COR nor DDO but appears though infrequently, e.g., in the title "Syv forundringer over resiliensbegrebet". COR records *vejr* (COR.44355) with plural forms while the compound *blæsevejr* (COR.66305) is without plural forms. DDO records them without plural forms and in Wikidata *vejr* has been labeled a singulare tantum.

At one point Danish nouns in Wikidata did not record the genitive forms. This was based on a discussion on the Danish *-s* as a clitic.[7] The Danish genitive *-s* can attach to phrases, e.g., even adverbs (Herslund, 2001), so if nouns should have genitive why not other lexemes from other lexical categories? Given that COR is representing nouns with genitive forms, we have started to add genitive forms for nouns in Wikidata.

Centaur nouns are often missing from dictionaries (Rajnik, 2009; Gregersen, 2014; Hansen and Heltoft, 2019). Many centaur nouns are missing in the core part of COR, but are listed in the COR-EXT, e.g., *søgen* and *banden* are not in the core part, but in COR-EXT. *indsynken* and *indsætten* used in medical texts ("indsynken i sig selv" and "akut indsætten") are centaur nouns that are missing in both resources. Other missing centaur nouns are *malen* and *truen*, both described in (Holm and Christensen, 2019). Centaur nouns are claimed to have no genitive form (Hansen and Heltoft, 2019, p. 612). Nevertheless COR-EXT records genitive forms, e.g., *søgens* and *bandens* and the rare genitive forms appear: An Internet search yields "denne søgens forløsning" and "denne søgens neutralitet".

5.4 Verbs

The initial entries of Danish verbs in Wikidata did not model the passive forms completely: The *-es* forms were annotated as one passive form. Following COR, we have now started to annotate the Danish verbs in Wikidata with two *-es* forms: The

[5] https://ordnet.dk/ddo/ordbog?query=drukning&search=Den+Danske+Ordbog

[6] A Google search with ""drukninger" site:dk"

[7] See https://www.wikidata.org/wiki/Wikidata_talk:Lexicographical_data/Archive/2018/10

passive–infinitive and passiv–present tense.

Verbs with multiple different conjugation are entered in Wikidata as different forms, e.g., *fise* has *fes*, *fiste* and *fisede* with the same temporal and grammatical features. Further Danish verbs that diverge from the normal 9-form conjugation scheme in Wikidata are deponent verbs that lack some forms and verbs ending with *-ere* where the imperative has an alternative spelling.

In COR 1.02, there are unusual *perfektum participium* forms for some of the common verbs, e.g., *hafte* (COR.30035.214.01) and *skullede* (COR.30128.214.01). They are not found in Retskrivningsordbogen nor in DDO.

5.5 Adjective

It has been unclear which grammatical features should be assigned to the different forms of Danish superlative (*-st* and *-ste*). Standard works in Danish grammar regard them as a kind of definite (or definite-ish) inflection (Diderichsen, 1962; Hansen and Heltoft, 2019). Danish Wiktionary[8] and the Swedish lexemes in Wikidata use the grammatical features predicative and attributive, see, e.g., *rolig* (L53287). COR 1.02 has 3 superlative forms, e.g., for *travl*: *travlest* (COR.15021.305.01, singular, indefinite), *travleste* (COR.15021.306.01, singular, definite) and *travleste* (COR.15021.307.01, plural). In Wikidata lexemes, we have only recorded two superlative forms: indefinite (*-st*) and definite (*-ste*).

COR includes comparative and superlative forms of adjectives that er quite rare, e.g., *radioaktivere*, *radioaktivest* and *radioaktiveste* from COR.26147 and *ugennemtænkere*, *ugennemtænktest*, *ugennemtænkteste* from COR.26148. With a simple Internet search, I was not able to find any examples of these forms, other than electronic dictionaries, while the periphrasic versions (e.g., *mere radioaktiv*) occur. Even *apropos* and *forleden*—which in the online version of Retskrivningsordbogen are regarded as uninflectable adjectives[9]—have comparative and superlative forms in COR 1.02.

Some of the superlative forms in COR are questionable: Adjectives that have no positive form, e.g., *ypperst* is specified with the positive form *ypperst*, but this should rather be superlative. COR has highly unusual *ypperstest* as the superlative form. Special cases are the compas adjectives *østre*, *nordre*, *vestre* og *søndre*. According to DDO they are originally comparative to *øst*, *nord*, *vest* og *syd*. COR regards these adjectives as in their positive form and records a comparative form, e.g., *nordrere*, and the superlative forms.

Nominalization of an adjective may result in a new lexeme noun in COR, e.g., the noun *indre* (COR.48793) is separate from the adjective and has a genitive form. These nominalizations are rare and adjectives do not have genitive in COR. Genitive forms have so far not been added to the adjectives in Wikidata.

6 Semantics

A few of the entries in COR have a short text for disambiguation of homographs. In a few cases it has been used as a gloss to the sense of a Wikidata lexeme, e.g., for *æg* COR disambiguates with "fx: fugleæg" (e.g., bird's egg) and "skarp kant" (sharp edge) for the two homographs and the Wikidata sense L39239-S1 "skarp kant" has been noted as a gloss referencing COR. For the other homograph, Wikidata has currently two senses (biological egg and egg as food) making the application of the disambiguation text as a sense gloss difficult. Such a case is not uncommon making automated setup or alignment of senses based the COR disambiguation text not feasible or at least difficult.

7 Discussion

There are Wikidata tools for mass-entry of lexemes and with COR data Danish Wikidata lexemes could be set up en masse. So far I have setup the links manually exploring the problems of ontology linking the two resources. I find *perfektum participium*, inflections of adjectives and nouns with both neuter and common gender are among the issues where one should be careful with matching. After the publication of COR, we have changed the entry of genitive for nouns and numerals and passive forms of verbs in Wikidata. I suspect that we might see a revision of inflections of adjectives in COR around comparative and superlative forms.

Acknowledgments

Thanks to Peter Juel Henrichsen and Thomas Widmann for discussions.

[8] See, e.g., https://da.wiktionary.org/wiki/ynkelig.

[9] E.g., https://dsn.dk/soegning/?soegeord=apropos

References

Paul Diderichsen. 1962. *Elementær Dansk Grammatik*. Gyldendal.

Christina Dideriksen, Peter Juel Henrichsen, and Thomas Widmann. 2022. Det Centrale Ordregister. *Nyt fra Sprognævnet* pages 2–6.

Sune Gregersen. 2014. "En hvislen i bækken". *Mål & mæle* pages 5–8.

Erik Hansen and Lars Heltoft. 2019. *Grammatik over det Danske Sprog*. Syddansk Universitetsforlag. https://www.universitypress.dk/shop/grammatik-over-det-3726p.html.

Michael Herslund. 2001. The Danish -s genitive: From affix to clitic. *Acta linguistica hafniensia* 33:7–18.

Lisa Holm and Robert Zola Christensen. 2019. *Dansk Grammatik*. Syddansk Universitetsforlag. https://www.universitypress.dk/shop/dansk-grammatik-3725p.html.

Finn Årup Nielsen. 2019. Danish in Wikidata lexemes. *Proceedings of the Tenth Global Wordnet Conference* pages 33–38.

Finn Årup Nielsen. 2020. Lexemes in Wikidata: 2020 status. *Proceedings of the 7th Workshop on Linked Data in Linguistics (LDL-2020)* pages 82–86.

Sanni Nimb, Bolette Sandford Pedersen, Nathalie Carmen Hau Sørensen, Ida Flörke, Sussi Olsen, and Thomas Troelsgård. 2022. COR-S – den semantiske del af Det Centrale OrdRegister (COR). *LexicoNordica* .

Bolette Sandford Pedersen, Sanni Nimb, Jørg Asmussen, Nicolai Hartvig Sørensen, Lars Trap-Jensen, and Henrik Lorentzen. 2009. DanNet: the challenge of compiling a wordnet for Danish by reusing a monolingual dictionary. *Language Resources and Evaluation* 43:269–299.

Bolette Sandford Pedersen, Nathalie Carmen Hau Sørensen, Sanni Nimb, Ida Flörke, Sussi Olsen, and Thomas Troelsgård. 2022. Compiling a Suitable Level of Sense Granularity in a Lexicon for AI Purposes: The Open Source COR Lexicon. *Proceedings of the Thirteenth Language Resources and Evaluation Conference* pages 51–60.

Eugeniusz Rajnik. 2009. Orddannelsesstrukturen af danske kentaurnominaler. *Folia Scandinavica Posnaniensia* 10:197–204.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM* 57:78–85.

Low-resource Bilingual Dialect Lexicon Induction with Large Language Models

Ekaterina Artemova **Barbara Plank**
MaiNLP, Center for Information and Language Processing (CIS), LMU Munich, Germany
Ekaterina.Artemova@lmu.de, B.Plank@lmu.de

Abstract

Bilingual word lexicons are crucial tools for multilingual natural language understanding and machine translation tasks, as they facilitate the mapping of words in one language to their synonyms in another language. To achieve this, numerous papers have explored bilingual lexicon induction (BLI) in high-resource scenarios, using a typical pipeline consisting of two unsupervised steps: bitext mining and word alignment, both of which rely on pre-trained large language models (LLMs).

In this paper, we present an analysis of the BLI pipeline for German and two of its dialects, Bavarian and Alemannic. This setup poses several unique challenges, including the scarcity of resources, the relatedness of the languages, and the lack of standardization in the orthography of dialects. To evaluate the BLI outputs, we analyze them with respect to word frequency and pairwise edit distance. Additionally, we release two evaluation datasets comprising 1,500 bilingual sentence pairs and 1,000 bilingual word pairs. They were manually judged for their semantic similarity for each Bavarian-German and Alemannic-German language pair.

1 Introduction

Learning in low-resource settings is one of the key research directions for modern natural language processing (NLP; Hedderich et al., 2021). The omnipresent pre-trained language models support high-resource languages by using increasingly large amounts of raw and labeled data. However, data scarcity hinders the training and evaluation of NLP models for less-resourced languages. At the same time, the participation of native speakers of different languages in the world of digital technologies increases the demand for supporting more language varieties. This encourages studies to explore suitable transfer learning and cross-lingual techniques.

Local varieties (dubbed as dialects) may fall under the umbrella of low-resource languages. Processing dialects faces unique challenges that should be addressed from a new perspective. Large volumes of writing in dialects such as newspapers or fiction are rarely produced and access to conversational data in social media is limited and difficult to reliably collect. Besides, dialects are non-standardized, they lack unified spelling rules and exhibit a high degree of variation (Millour and Fort, 2019). Finally, dialects may additionally show a significant rate of code-mixing to standard languages (Muysken et al., 2000).

The mainstream of cross-lingual transfer research towards low-resource languages, e.g., (Muller et al., 2021; Riabi et al., 2021), builds upon cross-lingual representations, namely static embeddings (Lample et al., 2018) or multilingual pre-trained language models (Devlin et al., 2019; Conneau et al., 2020). As shown by Muller et al. (2021) various factors can influence the performance, including the degree of relatedness to a pre-training language and the script. As there is no winning technique for all languages, it is important to understand how cross-lingual representations act for each particular language or a language family and whether the results in processing standard languages are transferable to its dialects.

In this paper, we focus on the ability of cross-lingual models to make semantic similarity judgments in German and two of its dialects, namely Bavarian (ISO 639-3:bar) spoken in South Germany, Austria, South Tyrol, and Alemannic (ISO 639-3:gsw) spoken in Switzerland, Swabia, parts of Tyrol, Liechtenstein, Alsace, and Italian re-

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 371–385

gions. Using the available raw data in Wikipedia (Section 3) we induce two bilingual lexicons, mapping words from Bavarian / Alemannic to German. To do so, we first mine bitext sentences (Section 4) and exploit machine translation aligners next (Section 5). The output lexicon exhibits an evident tendency for a German word to be aligned to multiple dialect synonyms due to spelling variations. Finally, we manually evaluate the output of each step: we evaluate the semantic similarity in (i) 1,500 bilingual sentence pairs according to the Likert scale and (ii) 1,000 bilingual word pairs according to a binary scale. Our results demonstrate the discrepancy between natural writing and linguistic dictionaries.

To sum up, this paper explores the following **research question** (RQ): How effective are standard pipelines for inducing bilingual lexicons for German dialects, and what factors influence their performance? To answer this question we make the following **contributions:** (i) We conduct a thorough analysis of cross-lingual models' behavior in two tasks of bitext mining and word alignment for the German language and two of its dialects. (ii) We release the evaluation datasets for bitext mining (1,500 samples each) and for bilingual lexicon induction (1,000 samples which). We make mined bitext dataset and induced bilingual lexicons for the Bavarian and Alemannic dialects publicly available. (iii) We publish the code to reproduce bitext extraction and word alignment in open access.[1]

2 Related work

NLP for German dialects. Previous efforts in processing German dialects mainly concentrate on speech processing. BAStat comprises the recordings of spoken conversational speech from main areas of spoken German (Schiel, 2010). Dogan-Schönberger et al. (2021) build a parallel corpus of spoken Alemannic dialect, in which a sentence in German is matched with spoken and written translations into eight dialects. ArchiMob is a general domain spoken corpus equipped with transcriptions and part-of-speech labeling (Scherrer et al., 2019). In the domain of written text processing, machine translation techniques have been applied to re-write sentences from dialect to standard German (Honnet et al., 2018; Plüss et al., 2020; Lambrecht et al., 2022). Other works tackle sentiment classification (Grubenmann et al., 2018), part-of-speech tagging (Hollenstein and Aepli, 2014) and dialect identification tasks (von Däniken et al., 2020). Burghardt et al. (2016) have collected a bilingual Bavarian-German lexicon using the knowledge of Facebook users, while Schmidt et al. (2020) hire expert native speakers to build a bilingual Alemannic-German lexicon. Language resources used to collect raw dialect data are Wikipedia, social media (Grubenmann et al., 2018), regional newspapers, and fiction (Hollenstein and Aepli, 2014). For a more comprehensive review, we refer to the concurrent survey of Blaschke et al. (2023).

Bitext mining. Sentence representations are core to mining bitext (dubbed as *parallel* or *comparable* datasets) in an unsupervised fashion (Hangya et al., 2018). Pires et al. (2019) show that `[CLS]`-pooling with multi-lingual encoders performs reasonably well for the task. Most recent studies proposed learning sentence embeddings from encoder-decoder models with a machine translation objective (Artetxe and Schwenk, 2019), by extending a monolingual sentence model to cross-lingual encoding with knowledge distillation (Reimers and Gurevych, 2020), or from dual encoders with a translation ranking loss (Feng et al., 2022). Bitext datasets collected from Wikipedia (Schwenk et al., 2021a) and the Common Crawl corpus (Schwenk et al., 2021b) serve to train machine translation models (Briakou et al., 2022) and to improve cross-lingual methods for structured prediction (El-Kishky et al., 2021). Chimoto and Bassett (2022) show that cross-lingual sentence models scale across unseen languages. We adopt the recent state-of-the-approach of (Reimers and Gurevych, 2020), which scores sentences embeddings, obtained from cross-lingual embedders, with cosine similarity measure in order to retrieve most similar sentence pairs.

Bilingual lexicon induction (BLI). Works in bilingual lexicon induction can be cast into two groups. *Mapping-based approaches* project monolingual word embeddings into a shared cross-lingual space with a varying degree of supervision (Lample et al., 2018; Artetxe et al., 2018; Joulin et al., 2018). *Corpora-based* methods combine bitext mining with word alignment (Shi et al., 2021). Intrinsic evaluation compares

[1] `https://github.com/mainlp/dialect-BLI`

Language	# dialects	# pages	# sent.	# tokens	#types	Manual labelling # bitext	Manual labelling # synonyms	Bilingual lexicon induction # bitext	Bilingual lexicon induction # synonyms
Bavarian	9	43k	230k	3.7mln	350k	1,254/1,500	860/1,000	17k	11k
Alemannic	32	71k	500k	9.5mln	600k	644/1,500	774/1,000	50k	194k
German	_____	3mln	56mln	106.4mln	1.12mln	_____	_____	_____	_____

Table 1: Left-hand part: Data statistics for Bavarian and Alemannic dialect Wikipedia. Alemannic Wikipedia is bigger than Bavarian, both are magnitudes smaller than standard German Wikipedia. Both Wikis label pages according to fine-grained dialects (# dialects). Center part: the number of sentence pairs manually labelled as similar (labels 4 and 5) out of 1,500 sentence pairs, the number of word pairs manually labelled as correct translations out of 1,000 word pairs. Right-hand part: the overall number of extracted bitext sentences, the overall number of extracted synonyms with a cutoff threshold of 0.8 for **MBERT** alignment probability.

induced bilingual lexicons to gold standard dictionaries (Rapp et al., 2020). Extrinsic evaluation is conducted through cross-lingual downstream tasks (Glavaš et al., 2019). Finally, several factor affect the quality of induced bilingual lexicons: edit distance, contextual and topical similarity between words in source and target languages (Scherrer, 2007; Irvine and Callison-Burch, 2017).

In this project, we apply best practices for *bitext mining* and *bilingual lexicon induction* and demonstrate their strengths and weaknesses in the low-resource settings of *German dialects*.

3 Data

Wikipedia offers articles written in more than 300 languages.[2] It is recognized that some parts of Wikipedia are human-translated (Schwenk et al., 2021a); examples are shown in (Table 2). The sentences for our bitext mining and bilingual lexicon induction experiments were extracted from Wikipedia pages in Bavarian[3], Alemannic[4], and German[5]. The Bavarian and Alemannic Wikipedias contain pages marked with nuanced variations in local dialects depending on the region of use. Of the nine dialects of the Bavarian Wikipedia, the most popular is *Westmittelbairisch* (Westmiddlebavarian), with nearly 3k pages. The Alemannic Wikipedia covers 32 dialect varieties, of which *Schwizerdütsch* (Swiss German) is the largest, containing 19k pages. In this work, we do not distinguish between these varieties and treat each Wikipedia as a single corpus.

We used the Wikipedia2corpus[6] tool to extract raw sentences. The texts were split into sentences and tokenized with the SoMaJo sentence splitter and tokenizer[7] (Proisl and Uhrig, 2016). The sentences were filtered by the 5-to-25 token range. Incomplete sentences were removed according to simple heuristics, such as the number of opening and closing brackets or the presence of a bullet point. Sentences containing non-German characters (e.g. letters from Greek, Cyrillic, and Hebrew alphabets) were filtered out. The left-hand part of Table 1 reports the total number of sentences, the number of tokens, and types per language in the resulting Wikipedia datasets. They illustrate the low-resource status of the Germanic dialects compared to the standard. The center part of Table 1 reports the size of manually labelled datasets for both tasks considered: bitext mining and bilingual lexicon induction in Bavarian and Alemannic. The right-hand site of Table 1 reports the sizes of automatically constructed datasets for both tasks in both dialects.

4 Bitext mining

Method. We start from the assumption parallel sentences are most often found on parallel pages, e.g. pages that are inter-linked between Wikipedias in two languages. We collect interlingual links between pages in dialect Wikipedia and German Wikipedia. Overall, we found 11k parallel pages for Bavarian and 32k parallel pages for Alemannic out of 43k and 71k, correspondingly. Given two parallel pages split into sentences, we embed each sentence with a language model. For each dialect sentence, we retrieve the nearest neighbors using the cosine similarity. Ta-

[2] `en.wikipedia.org/wiki/List_of_Wikipedias`, as of 01 Nov 2022

[3] `bar.wikipedia.org/`, as of 01 Nov 2022

[4] `als.wikipedia.org/`, as of 01 Nov 2022

[5] `de.wikipedia.org/`, as of 01 Nov 2022

[6] `github.com/GermanT5/wikipedia2corpus`

[7] `github.com/tsproisl/SoMaJo`

ble 2 provides examples of the found parallel sentences and the corresponding cosine similarity values. We aim to select the best-performing embedding model and the optimal cutoff value.

Models. We leverage the SentenceTransformer toolkit[8] (Reimers and Gurevych, 2020) for bitext mining. The experiments are with the following mono- and multi-lingual encoders and sentence models released as a part of HuggingFace library[9] (Wolf et al., 2020):

- **MBERT** (Devlin et al., 2019) was pre-trained on Wikipedia data. **MBERT** uses 110k shared across languages WordPiece vocabulary. Note that **MBERT** supports Bavarian and German.
- **GBERT** (Chan et al., 2020) was pre-trained on a range of different German language corpora. Training **GBERT** was carried out with the code-base used to train **MBERT**. Thus **GBERT** uses WordPiece tokenization. The size of vocabulary is 31k. Note that the exposure of **GBERT** to dialects is not mentioned explicitly.
- **GBERT-large-sts-v2**[10] is a version of **GBERT** fine-tuned the semantic textual similarity (STS) datasets of German sentence pairs.
- **LaBSE** (Feng et al., 2022) was pre-trained on the concatenation of mono-lingual Wikipedia and bilingual translation pairs. **LaBSE** uses the WordPiece tokenizer (Sennrich et al., 2016) trained with a cased vocabulary extracted from the model's training set. The vocabulary size is 501,153. **LaBSE** supports German but not its dialects.

We test both `[CLS]` and `[mean]` pooling[11] to obtain sentence representations from **MBERT** and **GBERT**. **GBERT-large-sts-v2** and **LaBSE** are sentence models and can be used out of the box to compute the similarity between sentences. **LaBSE** is the current state-of-the-art-model for bitext mining (Reimers and Gurevych, 2020).

Human evaluation. We sampled two random sets of 1,500 bitext instances with **LaBSE** similarity values in the $[0.4; 0.95]$ range to be manually labeled for semantic similarity and further justifications (see next and Appendix for details). We start from the LaBSE model since it is the current state-of-the-art model that has been shown to produce high-quality sentence embeddings (Ham and Kim, 2021). These embeddings capture both semantic and syntactic information, making them useful for a range of natural language processing tasks, including bitext mining. Furthemore, LaBSE was trained on a large-scale multilingual corpus, which makes it more robust and better able to handle variations in language and text structure (Feng et al., 2022).

The annotation schema utilized in our study is a five-point Likert scale, with a score of 5 indicating equivalence between the dialect sentence and the German sentence, and a score of 1 indicating no relation. Annotators were instructed to provide justifications for assigning scores that deviated from 5, by assessing the factual similarity between two given sentences, considering whether one sentence provided more information than the other. Additionally, annotators were asked to identify any significant differences in grammatical structure between the two sentences. The annotation instructions are provided in Section A. The Likert scale is a standardized approach to measuring sentence similarity, providing a more balanced set of response options when compared to binary judgments (Agirre et al., 2012). The annotations were carried out by a native German speaker with a linguistic background and significant exposure to dialects.[12] To ensure the quality of annotations, a smaller sample of 200 sentences was labeled by a second annotator, one of the authors, who is fluent in German.

The annotators were instructed to abstain from labeling sentence pairs with a Likert scale if they lacked a full understanding of the content, if the sentence was written in standard German rather than the dialect, or if the sentence contained a mixture of both. The inter-annotator agreement between the two annotators yielded a score of 0.80/0.78 for exact match and Pearson correlation, respectively, for Bavarian, and 0.9/0.6 for Alemannic. Notably, the primary source of confusion between the annotators was in labeling sentence pairs with scores that were in close proximity,

[8] `https://www.sbert.net`

[9] `https://huggingface.co`

[10] `https://hf.co/deepset/gbert-large-sts`

[11] The sentence representation obtained through `[CLS]` pooling uses the `[CLS]` token, while the sentence representation obtained through `[mean]` pooling averages token embeddings.

[12] The annotator was hired and received fair compensation according to the local employment regulations.

Bavarian	**German**	✎	COS
Da Geiselbach speist ob da Omersbachmindung oanige Weiher.	Der Geiselbach speist ab der Omersbachmündung einige Weiher.	5	0.94
Alemannic	**German**	✎	COS
Dr Film verzellt d'Geschichte vume Polizistepaar, dem si Idealismus im Lauf vu dr Handlig schwindet.	Der Film erzählt die Geschichte eines Polizistenpaares, deren Idealismus im Laufe der Handlung schwindet.	5	0.92

Table 2: Examples of parallel sentences in Bavarian and German (top) and Alemannic and German (bottom). ✎ denotes a human score (see Section 4 for more details on human evaluation), COS stands for the cosine similarity between **LaBSE** embeddings.

specifically (2,3) and (3,4). Notably, there were no instances in which the annotators disagreed and assigned opposite scores of 1 and 5. Additionally, the annotators were instructed to reject incomplete sentences or those not written in dialect, resulting in the rejection of 83 and 162 sentences, respectively. The remaining 1,417 and 1,338 sentence pairs in Bavarian and Alemannic were included for further analysis.

The results from the human annotation show that the distribution of labels is different for the two dialects: 1,254 sentences were labeled as similar (5) or near similar (4) for Bavarian and almost twice as less, 644 – for Alemannic. In the 250 Alemannic sentences marked with the label 3, the annotator pointed out that bitext sentences differ in minor factual details. Sentences in Bavarian differ less from their German counterparts, so that fewer than 100 sentences are marked as having differences in minor factual details. There are 250/350 Bavarian/Alemannic sentences labeled as using different grammatical structures such as active VS passive, imperfect VS perfect. In summary, based on this annotation study, we conclude that the authors of the Bavarian Wikipedia are more inclined towards literal translation, while the authors of the Alemannic Wikipedia rely less on translation.

Model comparison. Many of the retrieved sentence pairs have high similarity values. For instance, **LaBSE** assigns the scores of 0.8 or above to 42% and 24% of the dataset for Bavarian and Alemannic, correspondingly. Overall, the distribution of cosine values tends to be skewed to higher values for all embedders. **GBERT-large-sts-v2** shows the least reasonable performance: the average similarity value is 0.98 and the standard deviation is close to 0.01 for both dialects, leaving no discriminative power to select a precise cutoff threshold. This may happen due to over-fitting to semantic similarity tasks. The choice of pooling strategy does not affect the performance of **MBERT**: **MBERT+**`[CLS]` and **MBERT+**`[mean]` output strongly correlated cosine values (0.81 and 0.82 for Bavarian and Alemannic). This is not the case for **GBERT**, for which both `[CLS]` and `[mean]` pooling strategies lead to less correlated results ($\approx$0.5 for both dialects). We include the MBERT, GBERT, and LaBSE models in our comprehensive comparison of their abilities in bitext mining and bilingual lexicon induction (Section 5).

The resulting annotated dataset helps to evaluate whether the embedders can judge semantic similarity and assign lower scores to unrelated sentences. At the same time, we may use it to calibrate the cutoff threshold, which distinguishes between similar and unrelated sentence pairs. Figure 3 and Figure 4 in Appendix C show the cosine similarity values, derived with **MBERT+**`[CLS]`, **GBERT+**`[CLS]`, and **LaBSE** models, grouped according to Likert scale values. Although none of these models can divide the data into five groups, there is more evidence that **LaBSE** better separates unrelated sentences (scores 1, 2) from nearly similar or similar sentences (scores 4, 5) leaving somewhat similar sentences (score 3) in between. After careful consideration, we have chosen to use **LaBSE** in subsequent BLI experiments, setting the cutoff for the cosine similarity of nearly similar sentences to 0.7.

Results. Our bitext mining efforts resulted in 17k and 50k parallel Bavarian-German and Alemannic-German sentence pairs, respectively, sourced from Wikipedia. These pairs comprise 13.5% and 10% of the total number of sentences in their Wikipedia dumps, as shown in Table 1. After comparing various models, we have determined

that **MBERT** and **LaBSE** are the most closely aligned with human evaluation. This is likely due to **MBERT**'s previous exposure to dialect data, and **LaBSE**'s use of a sentence similarity objective during pre-training.

5 Bilingual lexicon induction

Method. We use the state-of-the-art awesome-align toolkit[13] (Dou and Neubig, 2021) with **MBERT** and **GBERT** as backbone models. Awesome-align supports an unsupervised mode, so there is no need to fine-tune the models on the parallel data. The word alignments are extracted from parallel sentences by evaluating the similarity between word representations. Awesome-align produces one-to-one alignment by default. When the source dialect sentence uses the perfect tense and the target German sentence uses the preterite tense, in the vast majority of cases, the auxiliary verbs align with the preterite verb.

We feed the extracted parallel dialect-German sentences to the aligner. The outputs are word pairs, in which one of the words is written in dialect and the other in German (see Table 3 for examples of word-level aligned parallel sentences). Each word pair is assigned with alignment probability (see Table 4 for sample output).

Next, we use several strategies to evaluate collected word pairs. We excluded word pairs that contained a non-word token, such as a number, typographical symbol, or punctuation mark. Previous research has demonstrated that the performance of BLI methods is highly dependent on word frequency, with higher frequency source words generally resulting in more accurate translations (Søgaard et al., 2018). To account for this, and to increase coverage of low-frequency words, we employed a stratified sampling approach for word selection in our evaluation. Specifically, we computed the frequency of each dialect word in Wikipedia and divided word pairs into four groups based on quartiles of dialect word frequency. From each group, we randomly selected 250 word pairs for further analysis.

Dictionary-based evaluation. To the best of our knowledge, there are no high-quality Bavarian-German or Alemannic-German lexicons, that can be easily accessed for computational experiments, so we turn to community-based resources. Glosbe[14] is a collection of community-maintained dictionaries, including Bavarian-German and Alemannic-German dictionaries. Since Glosbe does not provide an API, we manually look up German words and record the suggested translations into dialects.

Table 5 shows that the Glosbe dictionary provides better coverage for high-frequency words. The ratio of obtained translation sinks from 29% to 5% from high-frequency words to low-frequency words for Bavarian and from 26% to 4% for Alemannic. The low coverage of the Glosbe dictionary can be partially attributed to the absence of compounds, which are naturally present in Wikipedia writing. For instance, words such as *Laubwoidgebiet* (Bavarian, deciduous forest region) do not exist in Glosbe.

The mismatch between the induced translation and the dictionary-based translation is mainly caused by orthographic variations (see Table 6 for examples, in which both the induced and the Glosbe translations appear to be correct, but different from each other). This is especially evident in Alemannic, where only the 43% of high-frequency word pairs match to induced dictionaries.

Human evaluation. In addition to the dictionary-based evaluation, we also performed a human evaluation of the same word pairs using a binary scale to assess semantic similarity. Our aim was to determine whether a German word is a correct translation of a dialect word. The word pairs were presented without the surrounding context, and annotators were given the option to reject a word pair if they did not understand the dialect word. The use of a binary scale was chosen because it simplifies the assessment of semantic similarity and provides a clear indication of whether a word pair is a correct translation or not. The same annotators who participated in the evaluation of the bitext (Section 4) were recruited for the task. We provided annotators with guidelines that are detailed in Appendix B. To assess the level of agreement between annotators, we included a control sample consisting of 200 word pairs for each dialect. The exact match between annotators was high, with a score of 0.96 for Bavarian and 0.85 for Alemannic. Disagreements between annotators were mainly caused by judgments of overlapping words (*Turm – Kirchturm*, steeple – church steeple, in Bavarian).

[13] https://github.com/neulab/awesome-align

[14] https://glosbe.com/

Bavarian to German word alignment
Des Kloster Gunzenhausen is a obgongans Benediktinakloster im Bistum Eichstätt .
Das Kloster Gunzenhausen ist ein abgegangenes Benediktinerkloster im Bistum Eichstätt .

Alemannic to German word alignment
Um 1267 isch dr Heinrich I . Münch , dr Vater vom Hartung Münch , as Basler Bürgermäister erwähnt worde .
Um 1267 wurde __ Heinrich I . Münch , __ Vater von Hartung Münch , als Basler Bürgermeister erwähnt _____ .

Table 3: Examples of word level alignment in parallel sentences in Bavarian (top) / Alemannic (bottom) and German . Underscore ___ stands for unaligned words. **MBERT** is the backbone model.

Bavarian	**German**	#	P
Eihgmoant	Eingemeindet	112	0.99
Sidlichn	Südlichen	71	0.96
Augschburg	Augsburg	39	0.91
Alemannic	**German**	#	P
Dytsche	Deutsche	290	0.77
Yywohner	Bewohner	189	0.83
Uniwersidäät	Universität	126	0.95

Table 4: Examples of aligned word pairs in Bavarian (top) / Alemannic (bottom) and German. #: the frequency of the word pair. P stands for alignment probability. **MBERT** is the backbone model.

Q.	**Dictionary** ✍	**Dictionary** ✔	**Human** ✎
Bavarian: overall 860 words			
1	5%	50%	85%
2	6%	50%	95%
3	16%	65%	90%
4	29%	60%	65%
Alemannic: overall 774 words			
1	4%	70%	94%
2	5%	63%	95%
3	9%	81%	80%
4	26%	43%	40%

Table 5: Dictionary-based evaluation of induced bilingual lexicons, created from sentences, aligned with **LaBSE** and **MBERT** used as the aligner's backbone model. The results are grouped by the frequency quartile of German words, with 1 representing the low-frequency bin and 4 representing the high-frequency bin. Each bin contains 250 words. The percentage of words found in the Glosbe dictionary is denoted by ✍, while ✔ represents the percentage of matched word pairs between the dictionary and induced lexicons. The percentage of word pairs labeled as correct in human evaluation is denoted by ✎.

The evaluation of BLI through human annotation is presented in Table 5. The results indicate that the alignment quality of low-frequency and mid-frequency words is high, with a range of 85% to 95% in both dialects. However, for high-frequency words, the alignment quality drops significantly to 65% in Bavarian and 40% in Alemannic. This decline can be attributed to a higher prevalence of high-frequency prepositions, pronouns, and forms of auxiliary verbs that are often misaligned. Additionally, high-frequency words may contain multiple different spellings of the same word, leading to further noise in the alignment. This effect is more pronounced in Alemannic, where the number of fine-grained dialects is higher compared to Bavarian (as evidenced by Table 1 and the examples in Figure 1).

Interestingly, for mid-frequency words, one of the main sources of errors is the alignment of words that may be used in similar contexts but are not synonyms. For example, the word pair "Soizsään – Mineralquellen" in Bavarian, which translates to "salt lakes - mineral springs" in German, was found to be misaligned. Overall, the annotation study identified 860 and 774 out of 1,000 synonym word pairs between Bavarian and German, and Alemannic and German, respectively.

Baseline. We use supervised **MUSE** embeddings (Lample et al., 2018) as a baseline for bilingual lexicon induction. We employ fasttext embeddings, pre-trained on mono-lingual Wikipedia (Bojanowski et al., 2017), with identical character words as seeds. Following the project's guidelines,[15] we set up the dialect embeddings as the

[15] https://github.com/facebookresearch/MUSE

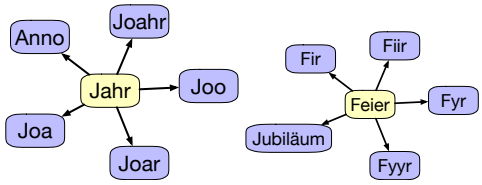


Figure 1: Manually picked examples of one-to-many correspondence from Bavarian-German (left) and Alemannic-German (right) bilingual lexicons. German words are in yellow, dialect words are in blue.

Bavarian	**German**	**Glosbe** $_{(de \rightarrow bar)}$
Obapfäjza	Oberpfälzer	Obapfejza
Zamm	Zusammen	Z'samm, zaum
Bavarian	**German**	**MUSE**$_{(de \rightarrow bar)}$
Vagressade	Vergrößerte	Großhadern
Freizeidzentrum	Freizeitzentrum	Sportpark
Alemannic	**German**	**Glosbe**$_{(de \rightarrow als)}$
Barlemäntarischi	Parlamentarische	Parlamentarischi
Nobelprys	Nobelpreis	Nobelpreis
Alemannic	**German**	**MUSE**$_{(de \rightarrow als)}$
Flüssige	Flüssiger	Wassermolekül
Epos	Heldengedicht	Heldenepos

Table 6: Differences between word pairs induced with **MBERT** and the Glosbe dictionary, **MUSE** synonyms.

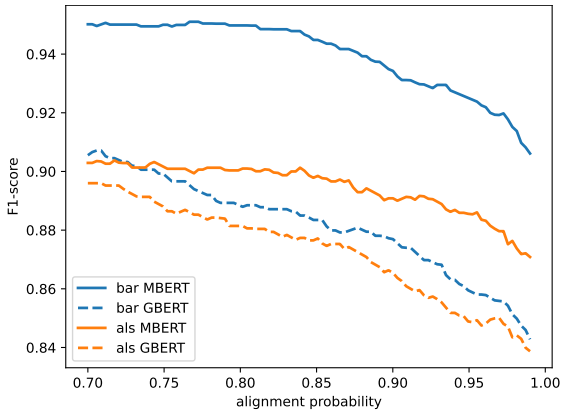


Figure 2: Comparison of two backbone models for Bavarian (blue) and Alemannic (orange). X axis: the cut-off threshold for alignment probability. Y axis: F_1 scores. The solid line stands for **MBERT**, and the dashed line stands for **GBERT**. **MBERT** consistently outperforms **GBERT** for both dialects.

source space and German embeddings as the target space. For each dialect word, we retrieve the nearest neighbor according to cosine similarity.

The **MUSE** embeddings retrieve 48 (out of 860) and 74 (out of 774) word pairs (Bavarian/Alemannic), identified as correct translations in the annotation study. Table 6 shows examples of cases, in which **MUSE** embeddings induce words that are different from those induced from bitext. These words have a similar spelling or can be used in similar contexts, but are not synonyms of source dialect words. Note, that the two-step approach for bitext mining and bilingual lexicon induction and the baseline **MUSE** embeddings leverage upon the same data source, namely, Wikipedia. However, our two-step approach leads to inducing more literal synonyms due to accessing larger contexts.

Model comparison. We conducted a comparison of two backbone models for the awesome-align toolkit in the binary classification setup. Specifically, we varied the threshold on alignment probability within the range of $[0.7; 0.99]$ and classified word pairs according to whether their probability was above or below the threshold. Negative and positive labels were assigned accordingly. We then compared these predictions to human yes/no labels and computed the F_1 score. The results are depicted in Figure 2.

Based on our analysis, it appears that the performance of **MBERT** reaches a plateau within the threshold range of $[0.7; 0.8]$ and gradually decreases as the threshold increases beyond this range. As a result, setting the cut-off threshold at 0.8 represents a reasonable choice. Furthermore, our results suggest that **MBERT** consistently outperforms **GBERT**. The superior performance of **MBERT** may be attributed to several factors, such as the inclusion of dialect data in its pre-training or the larger size of its tokenizer vocabulary.

Edit distance. Following prior works (Hangya et al., 2018), we explore the contribution of the edit distance to the word alignment probability. We compute the edit distance and normalize it with the sum of the number of characters in two words divided by two. The correlation coefficient

between the normalized edit distance and the average alignment probability makes -0.4 / -0.38 and $-0.49/-0.56$ for Bavarian with **MBERT / GBERT** backbones and Alemannic, respectively. This means, first, that the words, spelled similarly have higher chances to be aligned. Second, both backbone models significantly rely on the surface-level similarity between words. In our evaluation, edit distance was utilized solely for the purpose of assessment and not as a baseline. Despite its widespread use, edit distance is computationally expensive and is limited in its ability to capture semantic similarities. In lieu of this, we conducted a comparative analysis with MUSE embeddings, which take into account both surface and semantic similarity to provide a more comprehensive evaluation of the performance of our pipeline.

Results. After applying a cutoff threshold of 0.8 for **MBERT** alignment probability, we obtained bilingual lexicons containing 15,000 and 68,000 word pairs for Bavarian and Alemannic, respectively, as summarized in Table 1.

However, the resulting lexicons suffer from a high degree of word form repetition, as multiple dialect spellings are often linked to a single German word (see 1 for an illustrative example). Unfortunately, we were unable to merge different forms of the same word due to the lack of dialect stemmers, lemmatizers, or phonemizers. Words of different parts of speech were sometimes aligned, and we were unable to control for part of speech consistency due to the absence of dialect taggers. Clustering similar word forms presents an interesting avenue for future research.

6 Conclusion and Future Work

The project developed a two-stage pipeline for inducing bilingual lexicons for German and its dialects, Bavarian and Alemannic. The first stage involved extracting parallel sentences from public data, specifically Wikipedia, while the second stage used an alignment tool to induce word pairs from these parallel sentences. Both stages relied heavily on pre-trained LLMs, which were calibrated based on the results of annotation studies that judged the semantic similarity between extracted sentences and induced word pairs.

Returning to the research question raised, we may conclude that existing LLMs have a certain capacity for inducing bilingual lexicons. Our results have identified two key factors that influence their performance: (i) whether the pre-training included multilingual or dialect data, and (ii) whether the model was trained with a task-specific objective. Our evaluations demonstrate that the German GBERT is surpassed in both tasks, indicating that its monolingual pre-training is insufficient to effectively process related dialects. However, the main conundrum that remains is developing linguistic pipelines to process diverse and non-standardized dialect data. The development of dialect-specific tools such as lemmatizers, taggers and phonemizers can help improve the accuracy and consistency of bilingual lexicon induction.

Future work includes exploring the effect of fine-tuning cross-lingual LLMs on German and dialect data for bilingual lexicon induction, differentiating between several Bavarian/Alemannic dialects, and extending the experiments to other German dialects.

Limitations. While our study provides a comprehensive evaluation of induced bilingual lexicons for the Bavarian-German and Alemannic-German language pairs, there are some limitations to our approach. These limitations come with the low-resource setup.

Single domain. There is no large-scale dialect data source available, so we stick to Wikipedia as almost the only reasonable domain.

No extrinsic evaluation. One limitation is the lack of extrinsic evaluation due to the absence of annotated downstream datasets for these language pairs. We relied solely on intrinsic evaluation methods, which limits our ability to assess the usefulness of the induced lexicons in practical settings.

No multi-word expressions. Our evaluation focused on the alignment of individual words rather than multi-word expressions (MWEs).

Overall, the two-step pipeline of bitext mining and word aligning has its own disadvantages, such as resulting in one-to-one sentence / word alignment and over-relying on surface-level features.

Acknowledgements

Thanks to Anna Barwig for her contribution to the project on early stages. Special thanks to members of the MaiNLP lab for their feedback on this paper. This research is supported by ERC Consolidator Grant DIALECT 101043235.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Verena Blaschke, Hinrich Schütze, and Barbara Plank. 2023. A Survey of Corpora for Germanic Low-Resource Languages and Dialects. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Eleftheria Briakou, Sida Wang, Luke Zettlemoyer, and Marjan Ghazvininejad. 2022. BitextEdit: Automatic bitext editing for improved low-resource machine translation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1469–1485, Seattle, United States. Association for Computational Linguistics.

Manuel Burghardt, Daniel Granvogl, and Christian Wolff. 2016. Creating a lexicon of Bavarian dialect by means of Facebook language data and crowdsourcing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2029–2033, Portorož, Slovenia. European Language Resources Association (ELRA).

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Everlyn Chimoto and Bruce Bassett. 2022. Very low resource sentence alignment: Luhya and Swahili. In *Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2022)*, pages 1–8, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Pius von Däniken, Manuela Hürlimann, and Mark Cieliebak. 2020. Overview of the GermEval 2020 Shared Task on Swiss German Language Identification. In *5th SwissText & 16th KONVENS Joint Conference, Zurich (online), 24-25 June 2020*. CEUR Workshop Proceedings.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Pelin Dogan-Schönberger, Julian Mäder, and Thomas Hofmann. 2021. SwissDial: Parallel Multidialectal Corpus of Spoken Swiss German. *arXiv preprint arXiv:2103.11401*.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Ahmed El-Kishky, Adithya Renduchintala, James Cross, Francisco Guzmán, and Philipp Koehn. 2021. XLEnt: Mining a large cross-lingual entity dataset with lexical-semantic-phonetic word alignment. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10424–10430, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines,

comparative analyses, and some misconceptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 710–721, Florence, Italy. Association for Computational Linguistics.

Ralf Grubenmann, Don Tuggener, Pius von Däniken, Jan Deriu, and Mark Cieliebak. 2018. SB-CH: A Swiss German corpus with sentiment annotations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jiyeon Ham and Eun-Sol Kim. 2021. Semantic alignment with calibrated similarity for multilingual sentence embedding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1781–1791, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Viktor Hangya, Fabienne Braune, Yuliya Kalasouskaya, and Alexander Fraser. 2018. Unsupervised parallel sentence extraction from comparable corpora. In *Proceedings of the 15th International Conference on Spoken Language Translation*, pages 7–13, Brussels. International Conference on Spoken Language Translation.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.

Nora Hollenstein and Noëmi Aepli. 2014. Compilation of a Swiss German dialect corpus and its application to PoS tagging. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 85–94, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Pierre-Edouard Honnet, Andrei Popescu-Belis, Claudiu Musat, and Michael Baeriswyl. 2018. Machine translation of low-resource spoken dialects: Strategies for normalizing Swiss German. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ann Irvine and Chris Callison-Burch. 2017. A comprehensive analysis of bilingual lexicon induction. *Computational Linguistics*, 43(2):273–310.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium. Association for Computational Linguistics.

Louisa Lambrecht, Felix Schneider, and Alexander Waibel. 2022. Machine translation from standard German to alemannic dialects. In *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 129–136, Marseille, France. European Language Resources Association.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word Translation without Parallel Data. In *International Conference on Learning Representations*.

Alice Millour and Karën Fort. 2019. Unsupervised data augmentation for less-resourced languages with no standardized spelling. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 776–784, Varna, Bulgaria. INCOMA Ltd.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Pieter Muysken et al. 2000. *Bilingual speech: a Typology of Code-mixing*. Cambridge University Press.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Michel Plüss, Lukas Neukom, and Manfred Vogel. 2020. GermEval 2020 Task 4: Low-Resource Speech-to-Text. In *SwissText/KONVENS*.

Thomas Proisl and Peter Uhrig. 2016. SoMaJo: State-of-the-art tokenization for German web and social media texts. In *Proceedings of the 10th Web as Corpus Workshop*, pages 57–62, Berlin. Association for Computational Linguistics.

Reinhard Rapp, Pierre Zweigenbaum, and Serge Sharoff. 2020. Overview of the fourth BUCC shared task: Bilingual dictionary induction from comparable corpora. In *Proceedings of the 13th Workshop on Building and Using Comparable Corpora*, pages 6–13, Marseille, France. European Language Resources Association.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Arij Riabi, Benoît Sagot, and Djamé Seddah. 2021. Can character-based language models improve downstream task performances in low-resource and noisy language scenarios? In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 423–436, Online. Association for Computational Linguistics.

Yves Scherrer. 2007. Adaptive string distance measures for bilingual dialect lexicon induction. In *Proceedings of the ACL 2007 Student Research Workshop*, pages 55–60, Prague, Czech Republic. Association for Computational Linguistics.

Yves Scherrer, Tanja Samardžić, and Elvira Glaser. 2019. Digitising Swiss German: how to Process and Study a polycentric Spoken Language. *Language Resources and Evaluation*, 53(4):735–769.

Florian Schiel. 2010. BAStat : New statistical resources at the Bavarian archive for speech signals. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Larissa Schmidt, Lucy Linder, Sandra Djambazovska, Alexandros Lazaridis, Tanja Samardžić, and Claudiu Musat. 2020. A Swiss German dictionary: Variation in speech and writing. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2720–2725, Marseille, France. European Language Resources Association.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021a. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021b. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Haoyue Shi, Luke Zettlemoyer, and Sida I. Wang. 2021. Bilingual lexicon induction via unsupervised bitext construction and word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 813–826, Online. Association for Computational Linguistics.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788, Melbourne, Australia. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Bitext Annotation. Are these two sentences similar?

Task. Compare two sentences. One sentence is written in a Bavarian dialect. Another sentence is written in the standard German language. Your task is to compare these two sentences and decide how similar or different they are. You will be asked questions about sentence meaning, if one sentence provides more information than the other, and if the dialect sentence can potentially be a translated version of the German sentence.
Meaning. On a scale from 1 to 5, rate how close the meaning of sentences is. Choose the "idk" option if you do not understand the sentence to judge the similarity and skip the rest of the questions. Choose "n/a" if the first sentence is not in a Bavarian dialect and skip the rest of the questions. Choose "incomplete" if the dialect sentence is not complete or some parts of the sentence are missing and skip the rest of the questions. The scores can be interpreted in the following way.

Label	Explanation
idk	I do not understand the dialect sentence.
n/a	The dialect sentence is not written in the dialect.
incomplete	The dialect sentence is not complete (see below).
1	The sentences are completely unrelated.
2	The sentences have minor details in common (shared generic topic).
3	The sentences refer to same entities, but there are major differences (shared specific topic).
4	The sentences refer to same entities, but there are minor differences.
5	The sentences have identical meaning.

Table 7: The markup schema for bitext annotation.

Try to judge the differences between the sentences from the context. Do you learn the same things from these sentences or not? If one sentence adds more information, is it something really important?
Incomplete sentences. might look like these in Table 8 and should be labelled with 5.

Sentence	Is it complete?
Bédouès, Cocurès, Florac, Fraissinet-de-Lozère, La Salle-Prunet, Le Pont-de-Montvert, Saint-Andéol-de-Clerguemort, Saint-Frézal-de-Ventalon, Saint-Julien-d'Arpaon.	**No.** Reason: This looks like a part of a list.
House" Haus und des "Mordecai Lincoln House" Haus san historische Gebaide in Springfield und im National Register of Historic Places aufgfiaht.	**No.** Reason: It looks like a few words in the beginning of the sentence are missing.

Table 8: Examples of incomplete sentences.

Identical meaning. We consider sentences like these in Table 2 to have identical meaning.

Bavarian	German	Label
Da Geiselbach speist ob da Omersbachmindung oanige Weiher.	Der Geiselbach speist ab der Omersbachmündung einige Weiher.	5
Am 31. Dezemba 1990 werd Schladerlmühle ois unbewohnt und in Treffelstein aufgonga bezeichnt.	Am 31. Dezember 1990 wird Schladerlmühle als unbewohnt und in Treffelstein aufgegangen bezeichnet.	5
As Gebiet vo da Metropolitanstod Neapel is a bliabds Reisezui vo in- und ausländischn Touristn.	Das Gebiet der Metropolitanstadt Neapel ist ein beliebtes Reiseziel in- und ausländischer Touristen.	5

Table 9: Examples of sentences with identical meaning

Factual similarity. If the sentences do not have identical meaning, choose from the drop down list one of the following explanations, how they differ:

- The dialect sentence misses details.
- The dialect sentence adds details.

Bavarian	German	Meaning	Factual similarity
Seitm 1. Mai 2008 isa Easchta Buagamoasta vo da Gmoa Hafenlohr.	Seit dem 1. Mai 2008 ist er Erster Bürgermeister der Gemeinde Hafenlohr und Kreisrat im Landkreis Main-Spessart.	4	The dialect sentence misses details. Reason: The standard German sentence provides additional information (Kreisrat im Landkreis Main-Spessart).
De Gmoa eastreckt se iwa uma 54km².	Die Gemeinde erstreckt sich über etwa 55km².	4	Minor factual inconsistency. Reason: 54km² does not equal to 55km², but the numbers are almost the same.
Hafenlohr is duach des Zwoate Gmoaedikt am 17. Mai 1818 a Tei vo da Gmoa Hafenlohr gewoadn.	Hafenlohr ist der Hauptort der Gemeinde Hafenlohr.	3	Major factual inconsistency. Reason 1: The dialect sentence provides additional information (duach des Zwoate Gmoaedikt am 17. Mai 1818). Reason 2: The dialect sentence uses ”Tei”. The standard German sentence uses “Hauptort”.
As Spuin hod a recht wichtige Funktion in da Entwicklung vo Menschnkinda.	Von besonderer Bedeutung ist hier der Verlauf der individuellen Wachstumskurve.	2	Minor common details. Factual similarity is not applicable.
A Klassika vo humoristischa Litratua is aa da Ignát Herrmann.	Sein Werk ist ein lyrisches Poem, das hochromantisch und hochdramatisch ist.	1	Unrelated sentences Factual similarity is not applicable.

Table 10: Examples of sentence with scores from 1 to 4.

- Different details: both sentences add some new details and miss different details.
- Minor factual inconsistency.
- Major factual inconsistency.
- n/a: if the sentences are completely unrelated, you can not make a judgment about their factual consistency.

Table 10 shows instructions on how to score less similar sentences.

Grammar differs? This is a checkbox: mark yes, if there is a difference between the grammar structure of two sentences. If there is no difference or you can not tell it, skip this section.

Free form comment. Is there anything else you would like to notice about these two sentences? Can you explain the reason behind your judgment?

B Bilingual Lexicon Annotation. Is the translation acceptable?

Task. This project aims to evaluate bilingual word pairs. Each word pair consists of:

a. a word in Bavarian;
b. a word in Standard German.

The task is to label each word pair as an acceptable translation from Standard German to Bavarian. Label each word pair with:

1. **yes**, if the translation is acceptable;
2. **no**, if it is not acceptable;
3. **idk**, if you can not tell;
4. **check in the box “different part of speech”**, if the two words belong to different parts-of-speech only if you are sure you can tell it without full context.
5. **check in the box “partial match”**, if the words partially match and one word is a part of another (e.g. *Turm - Kirchturm, Sässl – Bürosessel).*

Free form comment. Put down free-form comments when necessary.

C Comparison of models for measuring sentence similarity

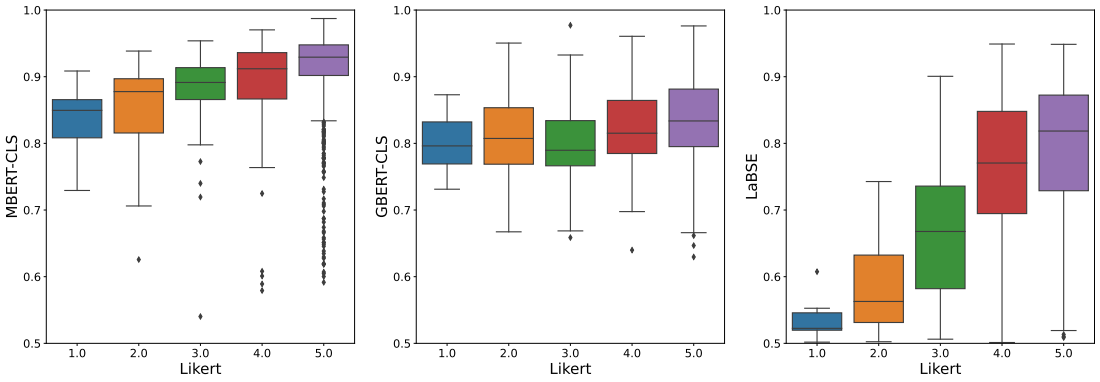


Figure 3: X axis: human scores in sentence similarity for Bavarian. Y axis: Cosine similarity values. The overall number of annotated sentences is 1,417.
Left: **MBERT** + `[CLS]`, middle: **GBERT** + `[CLS]`, right: **LaBSE**. The gap between unrelated and similar sentences is the most evident using **LaBSE** model.

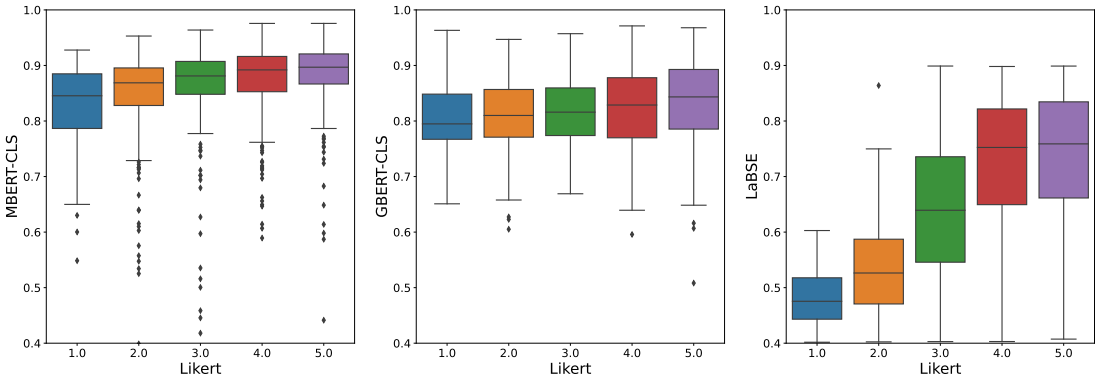


Figure 4: X axis: human scores in sentence similarity for Alemannic. Y axis: Cosine similarity values. The overall number of annotated sentences is 1,338.
Left: **MBERT** + `[CLS]`, middle: **GBERT** + `[CLS]`, right: **LaBSE**. The gap between unrelated and similar sentences is the most evident using **LaBSE** model.

Constructing a Knowledge Graph from Textual Descriptions of Software Vulnerabilities in the National Vulnerability Database

Anders Mølmen Høst
Simula Research Laboratory
& University of Oslo
Oslo, Norway
andersmh@simula.no

Pierre Lison
Norwegian Computing Center
& University of Oslo
Oslo, Norway
plison@nr.no

Leon Moonen
Simula Research Laboratory
& BI Norwegian Business School
Oslo, Norway
leon.moonen@computer.org

Abstract

Knowledge graphs have shown promise for several cybersecurity tasks, such as vulnerability assessment and threat analysis. In this work, we present a new method for constructing a vulnerability knowledge graph from information in the National Vulnerability Database (NVD). Our approach combines named entity recognition (NER), relation extraction (RE), and entity prediction using a combination of neural models, heuristic rules, and knowledge graph embeddings. We demonstrate how our method helps to fix missing entities in knowledge graphs used for cybersecurity and evaluate the performance.

1 Introduction

An increasing number of services are moving to digital platforms. The software used on these digital platforms is, unfortunately, not without flaws. Some of these flaws can be categorized as security vulnerabilities that an attacker can exploit, potentially leading to financial damage or loss of sensitive data for the affected victims. The National Vulnerability Database (NVD)[1] is a database of known vulnerabilities which, as of January 2023, contains more than 200 000 vulnerability records. The Common Vulnerability and Exposures (CVE) program[2] catalogs publicly disclosed vulnerabilities with an ID number, vulnerability description, and links to advisories. NVD fetches the data from CVE and provides additional metadata such as weakness type (CWE) and products (CPE). CWEs are classes of vulnerabilities (CVEs), for example, CWE-862: Missing Authorization contains all CVEs related to users accessing resources without proper authorization. A CPE is a URI string specifying the product and its version, for example, *cpe:2.3:a:limesurvey:limesurvey:5.4.15* is the CPE for the survey app Limesurvey with version 5.4.15. Keeping the information in the database up to date is important to patch vulnerabilities in a timely manner. Unfortunately, patching becomes increasingly difficult as the yearly number of published vulnerabilities increases.[3]

To automatically extract relevant information from vulnerability descriptions, named entity recognition (NER) and relation extraction (RE) can be applied as shown in Fig. 1. The extracted information can be stored as triples in a knowledge graph (KG). As the extracted triples might be incorrect or missing, knowledge graph embeddings (KGE) can be used to learn the latent structures of the graph and predict missing entities or relations.

The work described in this paper is based on the master thesis by the first author. We investigate how NLP and KGs can be applied to vulnerability records to predict missing software entities. More specifically, we address the following research question: *RQ: Can our knowledge graph predict vulnerability weakness types and vulnerable products?* The contributions of this paper include: (1) An approach for extracting and assessing vulnerability data from NVD; (2) A vulnerability ontology for knowledge graph construction; (3) A rule-based relation extraction model.

2 Related Work

We distinguish the ensuing areas of related work:
Labeling: Labeled data may not always be available to train supervised learning models for tasks including NER and RE. To address this problem, distant supervision aims at proposing a set of labeling functions for the automatic labeling of data. Bridges et al. (2014) applied distant supervision using a cybersecurity corpus. Their ap-

[1] https://nvd.nist.gov/
[2] https://www.cve.org/
[3] https://nvd.nist.gov/general/nvd-dashboard

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 386–391

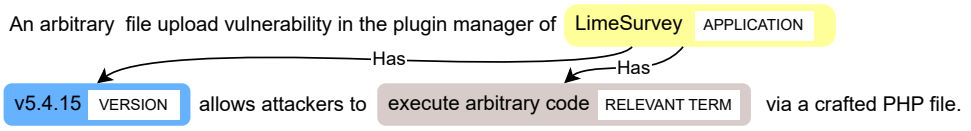


Figure 1: Example of a CVE with labels

proach includes database matching using the CPE vector, regular expressions to identify common phrases related to versioning, for example, "before 2.5", and *gazetteers*, which are dictionaries of vulnerability-relevant terms, such as "execute arbitrary code".

After manual validation of the labeled entities, Bridges et al. (2014) report a precision of 0.99 and a recall of 0.78.

Named Entity Recognition: Training NER models on labeled data are useful as distant supervision depends on assumptions about the input data, which does not always hold. For example, in the case of NVD, if the new data is missing CPE information. Machine learning models are not dependent on such metadata, and, as a consequence can generalize better to new situations. Bridges et al. (2014) propose NER based on the Averaged Perceptron (AP). The conventional perceptron updates its weights for every prediction, which can over-weight the final example. The averaged perception keeps a running weighted sum of the obtained feature weights through all training examples and iterations. The final weights are obtained by dividing the weighted sum by the number of iterations.

Gasmi et al. (2019) propose another NER model based on a long short-term memory (LSTM) architecture. The authors argue that it can be more useful when the data set has more variation, as the LSTM model does not require time-consuming feature engineering. However, their results show it is not able to reach the same level of performance as Bridges et al. (2014).

SecBERT[4] is a pre-trained encoder trained on a large corpus of cybersecurity texts. It is based on the BERT architecture (Devlin et al., 2019) and uses a vocabulary specialized for cybersecurity. SecBERT can be fine-tuned for specific tasks such as NER.

Another pre-trained encoder similar to SecBERT is SecureBERT, proposed by Aghaei et al. (2022). SecureBERT leverages a customized tokenizer and an approach to alter pre-trained weights. By altering pre-trained weights, SecureBERT aims to increase understanding of cyber security texts while reducing the emphasis on general English.

Relation Extraction: Relations between named entities can be discovered with RE. Gasmi et al. (2019) propose three RE models for vulnerability descriptions from NVD based on LSTMs. Their best-performing model achieves a precision score of 0.92. For labeling the relations, Gasmi et al. (2019), applies distant supervision (Jones et al., 2015). Gasmi et al. (2019) does not manually evaluate their labels before using them in the LSTM models; however, the approach is based on Jones et al. (2015), which indicates 0.82 in precision score after manual validation. Both NER and RE are important components for constructing knowledge graphs from textual descriptions. We explore several knowledge graphs related to cybersecurity in the next section.

Knowledge Graphs in Cybersecurity:

CTI-KG proposed by Rastogi et al. (2023), is a cybersecurity knowledge graph for Cyber Threat Intelligence (CTI). CTI-KG is constructed primarily from threat reports provided by security organizations, describing how threat actors operate, who they target, and the tools they apply. Rastogi et al. (2023) manually labels a data set of approximately 3000 triples with named entities and relations. This labeled data is then used for training models for NER and RE for constructing the KG. CTI-KG also uses KGE to learn latent structures of the graph and predict incomplete information.

Here, Rastogi et al. (2023) applies TuckER, a tensor decomposition approach proposed by Balažević et al. (2019), which can be employed for knowledge graph completion. TuckER can represent all relationship types (Balažević et al., 2019), as opposed to earlier models. For example, TransE proposed by Bordes et al. (2013) has issues modeling 1-to-n, n-to-1, and n-to-n relations (Lin et al., 2015). An example of a 1-to-n relationship in a cybersecurity context is the relationship between

[4]https://github.com/jackaduma/SecBERT

CVEs and CPEs. Whereas a CVE can have multiple CPEs, a CPE can only have one CVE.

As CTI-KG focuses on threats, another KG, VulKG (Qin and Chow, 2019), is constructed from vulnerability descriptions from NVD. VulKG consists of three components, a vulnerability ontology, NER for extracting entities from the vulnerability descriptions, and reasoning for discovering new weakness (CWE) chains. After extracting entities, relations between these can be found using the VulKG ontology (Qin and Chow, 2019). The final step of the framework presented by Qin and Chow (2019) is the reasoning component which is based on chain confidence for finding hidden relations in the graph.

Similarly to VulKG, we construct our KG from vulnerability descriptions in NVD. However, VulKG depends on training NER models from scratch, while we instead depend on a pre-trained model fine-tuned to our data. Contrary to training the model from scratch, the pre-training approach utilizes an existing model already trained on a large dataset. Consequently, fine-tuned models can learn patterns in the new data set more quickly.

3 Methods

Our approach is shown in Fig. 2 and gives an overview of the construction of the vulnerability knowledge graph from CVE records. We discuss the different steps below. For replication, we share details about the hyperparameter tuning of various models in the appendices.

Data: Our dataset is downloaded in JSON format from NVD, and the pipeline consists of multiple steps before predicting missing or incorrect labels as the final step. The data set consists of all CVE records from 2003 to 2022, which contains approximately 175 000 CVEs. The CVE records are labeled using the distant supervision approach proposed by Bridges et al. (2014).

Named Entity Recognition: We train two architectures: Averaged Perceptron and SecBERT.

Averaged Perceptron (AP): AP is a feature-engineered model, and we use the same features as Bridges et al. (2014) Due to computational constraints in the AP model, we restricted our training data to 4000 CVEs.

We first replicate their approach and separately trained and evaluated two AP models, one for IOB-labeling and one for domain-labeling, using the distant supervision-generated labels. In practice, when a new CVE is published, we only have access to the textual description. Since the IOB labels are input features to the domain model, those must be predicted first. Thus, in our second experiment, we again train two AP models, but use the predicted IOB labels as input to the domain labeling, instead of the generated labels.

SecBERT: In addition to AP, we use the pre-trained SecBERT model for NER. A significant difference from AP is that SecBERT jointly extracts IOB and domain labels. Moreover, as SecBERT is significantly faster than AP, there is no need to restrict the dataset. We split our data into 60/20/20 for training, evaluation, and testing.

Relation Extraction: For relation extracting, we use an *ontology* illustrated in Fig. 3, to guide their creation: When two entities of type A and B are detected in a CVE, a relation between the two is created if the ontology has an edge between types A and B.

Note that entities are connected to their corresponding CVE-ID and CWE-ID, and we concatenate multi-word entities based on their IOB labels.

The vulnerability descriptions are generally written so that vendors are followed by their products which are then followed by their versions. Thus, we can derive relations between vendor, product, and version by looking at the word order. We also make relations from relevant terms to the corresponding CVE ID entity, and through the CVE-ID the relevant terms are connected to the corresponding vendors, products, and versions.

Entity Prediction: To answer the RQ, our KG should predict weakness types (CWEs) and products (CPEs). Given a head entity and a relation as input, the task of entity prediction is to find the tail entity, which is the final step of our KG. $Hits@n$ and *mean reciprocal rank* (MRR) are standard metrics used for entity prediction. For each input example, the embedding algorithm assigns a confidence score to all possible triples. These triples are then ranked by confidence scores, where the triple with the highest confidence is the most plausible to be true according to the model. The $Hits@n$ metric measures the number of times the true triple is ranked among the top n triples. We use the processed triples from the RE model as input to our entity prediction model, where TuckER is the chosen architecture. The triples from our RE model are considered ground truth. TuckER removes the

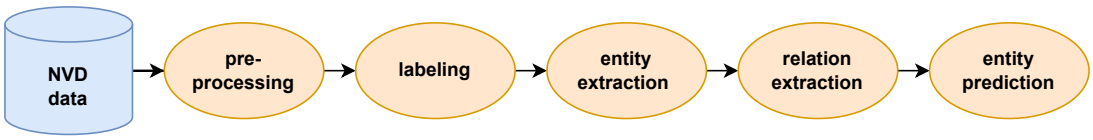


Figure 2: The figure illustrates the steps in our approach. We start by downloading our data from NVD, pre-processing the data, and adding labels to the entities. With our labeled data, we perform NER and RE to construct the KG. Because missing entities might occur in the KG, we predict these in the last step.

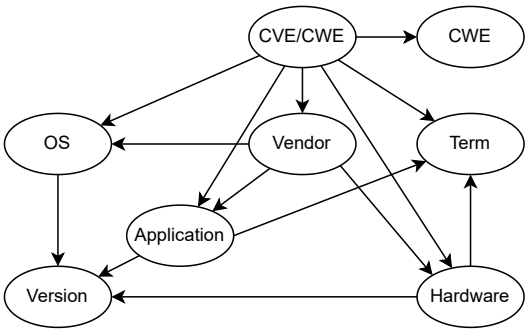


Figure 3: Ontology for relation extraction. The edges should be interpreted as, for example, "a vendor *has* a product", "a product *has* a version", "a CVE vulnerability *has* a CWE type"

tail entities from the ground truth before predicting these based on entity and relation embeddings. We perform data augmentation by reversing all the relational triples. The data set is split in 80/10/10 percent for training, validation, and testing. We select the best model by refining the four combinations proposed by Balažević et al. (2019) with an additional grid search.

4 Results and discussion

Our empirical evaluation uses the CVE dataset discussed in Section 3. For replication, the parameters of the best-performing models are in the appendices.

NER: NER results are presented in Table 1. We see that SecBERT outperforms AP on all metrics.

We compare our reproduction results with the results reported by Bridges et al. (2014) in Table. 2. Where Table 1 shows the performance with all labels in place, individual IOB and domain labeling performance are reported in Table 2. The AP model was based on Bridges et al. (2014), which implemented their experiments in OpenNLP and Python. We reused their Python code for our reproduction. Note that the results on our data are below the reports by Bridges et al.. The authors indicated that they experienced slightly better performance using OpenNLP, which *could* be the reason for the difference in score. Unfortunately, they do not provide any explanation of this difference or why it occurs. Contrary to Bridges et al. (2014), we are not interested in the performance of IOB and domain labeling measured individually. In our approach, the NER model should be used to extract entities from new data that can form triples in our KG. When a new CVE is published, we can access the textual description without any labels. Using Bridges' approach, we first need to use the IOB model, and then the predicted IOB labels can be used as input features to the domain model responsible for the final prediction.

Table 1: NER evaluation results for the averaged perception and the fine-tuned SecBERT model.

NER Model	Precision	Recall	F_1
Averaged perceptron	0.925	0.84	0.88
Fine-tuned SecBERT	0.93	0.93	0.93

Table 2: Our reproduction results compared to those reported by Bridges et al. (2014)

Author	Labeling	Precision	Recall	F_1
Høst et al.	IOB	0.93	0.93	0.93
	Domain	0.94	0.94	0.94
Bridges	IOB	0.97	0.97	0.96
	Domain	0.99	0.99	0.99

To the best of our knowledge, we can not analytically combine the IOB model and domain model results reported by Bridges et al.. As such, we rely on our own experimental results, which show that the performance of the fine-tuned SecBERT model outperforms the AP model.

Relation Extraction: We did not have any ground truth data when evaluating our RE approach, as a consequence, we manually validated

Table 3: Performance metrics for our entity prediction model compared to Rastogi et al. (2023).

Model	Hits@10	Hits@3	Hits@1	MRR
Høst et al.	0.760	0.728	0.682	0.710
Rastogi	0.804	0.759	0.739	0.75

a sample of 100 extracted triples. From this sample, we measured a precision score of 0.77. While Jones et al. (2015) has proposed a semi-supervised approach for labeling relations, they focus on a broader data set than we do. We, therefore, choose to identify relations based on our proposed ontology in Fig. 3. Our RE approach could not reach the level of Jones et al. (2015), which reported 0.82 in precision score. For future work, one idea to improve RE is to utilize CPE vectors for relation labeling in addition to our proposed rules. Then we can train machine learning models on top of our labeled data using pre-trained variations of BERT models.

Entity Prediction: During the relation extraction, we extracted approximately two million triples. As we further reversed all triples, four million triples were used as input to the model.

In Table. 3, we compare our best-performing model with the results presented in Rastogi et al. (2023), which uses the same model architecture, TuckER, on threat reports. The input data are assumed to be true, and evaluation performance is not manually validated.

We choose TuckER as our embedding algorithm for entity prediction as it is the current state-of-the-art model measured on standard data sets (Balažević et al., 2019). The idea is that TuckER captures latent structures of our KG. TuckER encodes the input triples as vector embeddings based on encoded characteristics and can use these embeddings to predict missing entities. For example, if two CVEs share important characteristics such as vulnerability-relevant terms and affected products, then according to the theory, they should belong to the same neighborhood in a vector space. Consequently, TuckER could predict that the CVEs belong to the same CWE.

$Hits@n$ and *mean reciprocal rank* (MRR) are standard metrics used for entity prediction. Given a head entity and a relation, the task is to predict the tail entity. For each example, the embedding algorithm assigns a confidence score to all possible triples. These triples are then ranked by confidence scores, where the triple with the highest confidence is the most plausible to be true according to the model. The $Hits@n$ metric measures the number of times the true triple is ranked among the top n triples.

As a benchmark to measure our performance, we use the results presented in Rastogi et al. (2023), which also uses TuckER for entity prediction. Rastogi et al. (2023) has reported a Hits@10 metric of 0.804, which is better than our reported results seen in Table 3. We believe that more precise and consistent input labels can be the reason for this, where a limitation of our approach is that we aim at predicting CVE-IDs which are unique for each vulnerability description. We consider the task of predicting CVE-IDs as less important for our model as these will always be attached to the CVE description from our raw data. Balažević et al. (2019) addresses that future work might incorporate background knowledge on relationship types. Avoiding predicting CVE-IDs is one example of such background knowledge.

Another reason for the difference could be that some CWEs overlap and share many of the same entities making it more difficult for our model to discriminate between CWEs.

5 Conclusion

This paper proposes a vulnerability knowledge graph constructed from textual CVE records from the National Vulnerability Database (NVD). The graph construction relies on a pipeline including NER, relation extraction, and an entity prediction model based on the TuckER framework.

As future improvements, we are interested in better labeling of relations through distant supervision approaches and the integration of BERT models for relation extraction.

References

Ehsan Aghaei, Xi Niu, Waseem Shadid, and Ehab Al-Shaer. 2022. SecureBERT: A Domain-Specific Language Model for Cybersecurity.

Ivana Balažević, Carl Allen, and Timothy M. Hospedales. 2019. TuckER: Tensor Factorization for Knowledge Graph Completion. In *Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5184–5193.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Robert A. Bridges, Corinne L. Jones, Michael D. Iannacone, Kelly M. Testa, and John R. Goodall. 2014. Automatic Labeling for Entity Extraction in Cyber Security. *arXiv:1308.4941 [cs]*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL, Vol 1)*, pages 4171–4186. Association for Computational Linguistics.

Houssem Gasmi, Jannik Laval, and Abdelaziz Bouras. 2019. Information Extraction of Cybersecurity Concepts: An LSTM Approach. *Applied Sciences*, 9(19):3945.

Corinne L. Jones, Robert A. Bridges, Kelly M. T. Huffer, and John R. Goodall. 2015. Towards a Relation Extraction Framework for Cyber-Security Concepts. In *Annual Cyber and Information Security Research Conference*, CISR '15, pages 1–4, New York, NY, USA. Association for Computing Machinery.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

Shengzhi Qin and K. P. Chow. 2019. Automatic Analysis and Reasoning Based on Vulnerability Knowledge Graph. In *Cyberspace Data and Intelligence, and Cyber-Living, Syndrome, and Health*, Communications in Computer and Information Science, pages 3–19, Singapore. Springer.

Nidhi Rastogi, Sharmishtha Dutta, Mohammed J. Zaki, Alex Gittens, and Charu Aggarwal. 2023. TINKER: A framework for Open source Cyberthreat Intelligence. *arXiv:2102.05571 [cs]*.

Appendices

A SecBERT NER Tuning

We first perform a grid search (four epochs) over the 20 parameter combinations recommended by the BERT authors.[5] The grid consisted of batch sizes: {8, 16, 32, 64, 128}, and learning rates: {3e-4, 1e-4, 5e-5, 3e-5}. The most promising candidates were then trained for ten epochs.

[5] https://github.com/google-research/bert

B Entity Prediction Tuning

For tuning hyperparameters, we follow two strategies: First, we train the same four combinations as was done by (Balažević et al., 2019). These four models were run for 100 epochs and based on the intermediate results, the most promising model was run for additional 200 epochs such that this model was trained for 300 epochs in total. We select the best-performing model and based on its characteristics set up an additional grid search covering 36 hyperparameter combinations on smaller subsets of the data. To avoid overfitting, two models were trained and evaluated for each of the hyperparameter combinations on different subsets. Our grid consisted of values of hidden dropouts: {0, 0.1, 0.2}, learning rates: {0.001, 0.01, 0.1} and dimensions: {10, 30, 200}. The parameters from the most promising candidate were used for training another model for 300 epochs on the full dataset.

C Best Model for NER

The best SecBERT model for NER was trained with a learning rate of 5e-5 and a batch size of 8.

D Best Model for Entity Prediction

The following are the hyperparameters of the best-performing TuckER model:

Model	TuckER
num_iterations	300
edim	200
rdim	30
lr	0.001
input_dropout	0.2
hidden_dropout1	0.1
hidden_dropout2	0
batch_size	128
label_smoothing	0.1
dr	1

A Survey of Corpora for Germanic Low-Resource Languages and Dialects

Verena Blaschke **Hinrich Schütze** **Barbara Plank**
Center for Information and Language Processing (CIS), LMU Munich, Germany
Munich Center for Machine Learning (MCML), Munich, Germany
blaschke@cis.lmu.de inquiries@cislmu.org bplank@cis.lmu.de

Abstract

Despite much progress in recent years, the vast majority of work in natural language processing (NLP) is on standard languages with many speakers. In this work, we instead focus on low-resource languages and in particular non-standardized low-resource languages. Even within branches of major language families, often considered well-researched, little is known about the extent and type of available resources and what the major NLP challenges are for these language varieties. The first step to address this situation is a systematic survey of available corpora (most importantly, annotated corpora, which are particularly valuable for NLP research). Focusing on Germanic low-resource language varieties, we provide such a survey in this paper. Except for geolocation (origin of speaker or document), we find that manually annotated linguistic resources are sparse and, if they exist, mostly cover morphosyntax. Despite this lack of resources, we observe that interest in this area is increasing: there is active development and a growing research community. To facilitate research, we make our overview of over 80 corpora publicly available.[1]

1 Introduction

The majority of current NLP today focuses on standard languages. Much work has been put forward in broadening the scope of NLP (Joshi et al., 2020), with long-term efforts pushing boundaries for language inclusion, for example in resource creation (e.g., Universal Dependencies (Zeman et al., 2022)) and cross-lingual transfer research (de Vries et al., 2022). However, even within major branches of language families or even single countries, plenty of language varieties are under-researched.

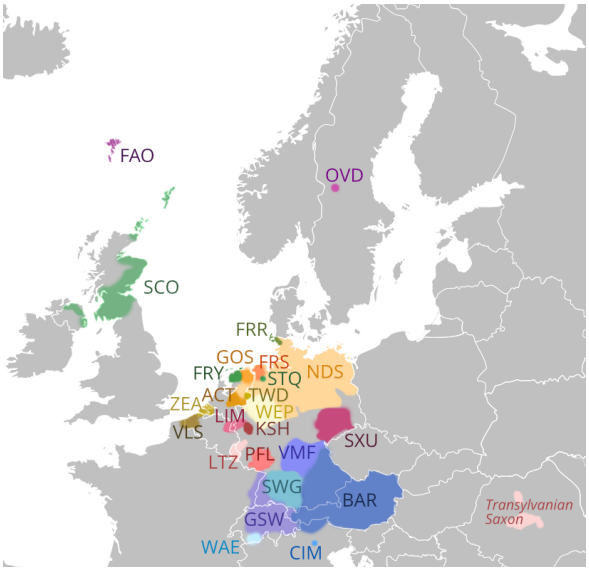


Figure 1: **Approximate locations of most of the languages discussed in this article** (not pictured: PDC, YID, NOR, SWE, DAN, ENG, DEU). Based on a map of Europe by Marian Sigler, CC BY-SA 3.0.

Current technology lacks methods to handle scarce data and the rich variability that comes with low-resource and non-standard languages. Nevertheless, interest in these under-resourced language varieties is growing. It is a topic of interest not only for (quantitative) dialectologists (Wieling and Nerbonne, 2015; Nerbonne et al., 2021), but also NLP researchers, as evidenced by specialized workshops like VarDial[2], special interest groups for endangered[3] and under-resourced languages,[4] and recent research on local languages spoken in Italy (Ramponi, 2022), Indonesia (Aji et al., 2022) and Australia (Bird, 2020), to name but a few.

[1] We share a companion website of this overview at github.com/mainlp/germanic-lrl-corpora.

[2] sites.google.com/view/vardial-2023

[3] SIGEL, acl-sigel.github.io

[4] SIGUL, www.elra.info/en/sig/sigul

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 392–414
May 22-24, 2023

In this paper, we provide an overview of the current state of NLP corpora for Germanic low-resource languages (LRLs) and dialects, with a particular focus on non-standard variants and four dimensions: annotation type, curation profile, resource size, and (written) data representation. We find that the amount and type of data varies by language, with manual annotations other than for morphosyntactic properties or the speaker's dialect or origin being especially rare. With this survey, we aim to support development of language technologies and quantitative dialectological analyses of Germanic low-resource languages and dialects, by making our results publicly available. Finally, based on the experiences we made while compiling this survey, we share recommendations for researchers releasing or using such datasets.

2 Related surveys

Zampieri et al. (2020) provide an overview on research on NLP for closely related language varieties and mention a few data sets. Recently, several surveys focusing on NLP tools and corpus linguistics data for regional languages and dialects have been released: for local languages in Italy (Ramponi, 2022) and France (Leixa et al., 2014), indigenous languages of the Americas (Mager et al., 2018), Arabic dialects (Shoufan and Alameri, 2015; Younes et al., 2020; Guellil et al., 2021), creole languages (Lent et al., 2022), Irish English (Vaughan and Clancy, 2016), and spoken varieties of Slavic languages (Dobrushina and Sokur, 2022). Furthermore, Bahr-Lamberti (2016) and Fischer and Limper (2019)[5] survey digital resources for studying varieties closely related to German, although these do not necessarily fit our inclusion criteria (cf. Section 4).

3 Language varieties

Our survey contains corpora for more than two dozen Germanic low-resource varieties, selected based on dataset availability (Appendix A contains the full list). We focus on specialized corpora showcasing regional variation, but not necessarily global variation. This overview does not include any corpora for Germanic-based creoles like Naija, as those are included in the recent survey by Lent et al. (2022). Figure 1 shows where most of the doculects included in this survey are spoken.

[5] regionalsprache.de/regionalsprachenforschung-online.aspx

4 Methodology

Similarly to Ramponi (2022), we search for corpora on several online repositories for language resources: the CLARIN Virtual Language Observatory (Van Uytvanck et al., 2010), the LRE Map (Calzolari et al., 2012), the European Language Grid (Rehm et al., 2020) OLAC (Simons and Bird, 2003), ORTOLANG (Pierrel et al., 2017), and the Hamburg Centre for Language Corpora.[6] We also search for corpora on Zenodo and on Google Dataset Search, and look for mentions of corpora in articles hosted by the ACL Anthology and on ArXiv.[7] We search for mentions of the word "dialect" and the names of various Germanic low-resource languages.

We use the following inclusion criteria:

- The corpus is accessible to researchers (immediately via a website, or indirectly through a request form or via contact information),[8] and this is indicated on the corpus website or in a publication accompanying the corpus.
- The corpus can be downloaded easily (does not require scraping a website) and does not require extensive pre-processing (we are interested in file formats like XML, TSV or TXT rather than PDF).
- The data are of a high quality, e.g., we ignore OCR'ed corpora that were not carefully cleaned.
- The corpus (mainly) contains full sentences or utterances,[9] and the data were (mainly) produced in the past century.

We base this survey only on the versions of corpora that are easily accessible to the research community; e.g., if a corpus contains audio material, but only the written material is available for download (and thus as a data source for quantitative research), the corpus is treated as a text corpus.[10]

[6] vlo.clarin.eu; lremap.elra.info; live.european-language-grid.eu; www.language-archives.org; www.ortolang.fr/market/corpora; corpora.uni-hamburg.de/hzsk/en/repository-search

[7] zenodo.org; datasetsearch.research.google.com; aclanthology.org; arxiv.org

[8] The latter case is indicated with a lock 🔒 in the tables.

[9] This excludes word lists and some heavily preprocessed corpora, like the one by Hovy and Purschke (2018), which is lemmatized and does not contain stop words.

[10] This is not a rare scenario, as the audio versions might

Corpus	Langs	Annotation	Size	Rep.
UD Faroese OFT (Tyers et al., 2018) `github.com/UniversalDependencies/UD_Faroese-OFT`	FAO	POS (UPOS, Giellatekno-FAO), dep (UD), morpho (UD), lemmas	1.2k sents	A
FarPaHC / UD Faroese FarPaHC (Ingason et al., 2012; Rögnvaldsson et al., 2012) `hdl.handle.net/20.500.12537/92` `github.com/UniversalDependencies/UD_Faroese-FarPaHC`	FAO	POS (mod. Penn-h, UPOS), phrase struc.(mod. Penn-h), dep (UD), morpho (UD)	53k (FarP.) / 40k (UD.) toks	A
LIA Treebank / UD Norwegian NynorskLIA (Øvrelid et al., 2018) `tekstlab.uio.no/LIA/norsk/index_english.html` `github.com/UniversalDependencies/UD_Norwegian-NynorskLIA` `github.com/textlab/spoken_norwegian_resources/tree/master/treebanks/Norwegian-NynorskLIA`	NOR 📍	POS (UPOS, mod. NDT), dep (UD, mod. NDT), lemmas, morpho (UD)	77.7k toks (L.), 55k toks (UD)	¶ ✎*
NDC Treebank (Kåsen et al., 2022; Johannessen et al., 2009) `tekstlab.uio.no/scandiasyn/download.html` `github.com/textlab/spoken_norwegian_resources/tree/master/treebanks/Norwegian-BokmaalNDC`	NOR 📍	POS (mod. NDT), dep (mod. NDT), lemmas, morpho (mod. NDT)	66k toks	¶ ✎*
NorDial (subset) (Mæhlum et al., 2022) Contact authors 🔒	NOR	POS (UPOS)	35+ tweets	✎
POS-tagged Scots corpus (Lameris and Stymne, 2021) `github.com/Hfkml/pos-tagged-scots-corpus`	SCO	POS (UPOS)	1k tokens	✎/A
TwitterAAE-UD (Blodgett et al., 2016) `slanglab.cs.umass.edu/TwitterAAE`	ENG (AAVE)	Dep (UD)	250 tweets	✎
UD Frisian/Dutch Fame (Braggaar and van der Goot, 2021; Yılmaz et al., 2016) `github.com/UniversalDependencies/UD_Frisian_Dutch-Fame`	FRY/NLD	POS (UPOS), dep (UD), code-switching	400 sents	A
UD Low Saxon LSDC (Siewert et al., 2021) `github.com/UniversalDependencies/UD_Low_Saxon-LSDC`	NDS 📍	POS (UPOS), dep (UD), morpho (UD), glosses (GML), lemmas	95 sents	✎ ¶*
Stemmen uit het verleden (annotated subset) (Lybaert et al., 2019; Van Keymeulen et al., 2019) `doi.org/10.18710/NSFN2B`	VLS 📍	V2 variation	1.4k sents	✎
Penn Parsed Corpus of Historical Yiddish (Santorini, 2021) `github.com/beatrice57/penn-parsed-corpus-of-historical-yiddish`	YID	POS (Penn-h), phrase struc. (Penn-h)	ca. 200k toks	*
Kontatto (Dal Negro and Ciccolone, 2020) `kontatti.projects.unibz.it` 🔒	BAR (South Tyrol)	POS (unknown), lemmas (DEU)	147k toks	🎙 ✎
Annotated Corpus for the Alsatian Dialects (Bernhard et al., 2018, 2019) `zenodo.org/record/2536041`	GSW (Alsatian)	POS (UPOS, mod. UPOS), lemmas, glosses (FRA)	798 sents	✎
Bisame GSW (STIH, 2020; Millour and Fort, 2018) `hdl.handle.net/11403/bisame_gsw/v1`	GSW (Alsatian)	POS (mod. UPOS)	382 sents	✎
Geparstes und grammatisch annotiertes Korpus schweizerdeutscher Spontansprachdaten (Schönenberger and Haeberli, 2019) (contact authors 🔒)	GSW (St. Gallen)	POS (mod. Penn-h), phrase struc. (Penn-h)	100k+ toks	🎙 ✎
NOAH's corpus (Hollenstein and Aepli, 2015) `noe-eva.github.io/NOAH-Corpus`	GSW	POS (mod. STTS, subset: UPOS and STTS)	115k toks	✎
UD Swiss German UZH (Aepli and Clematide, 2018) `github.com/UniversalDependencies/UD_Swiss_German-UZH`	GSW	POS (UPOS, mod. STTS), dep (UD)	100 sents	✎
WUS_DIALOG_GSW (subset of *What's up, Switzerland?*) (Stark et al., 2014–2020; Ueberwasser and Stark, 2017) `whatsup.linguistik.uzh.ch` 🔒	GSW 📍	POS (mod. STTS)	34.7k toks	✎ ¶

Table 1: **Morphosyntactically annotated corpora.** Abbreviations for the annotation tag sets are explained in Section 5.1.1, as are the orthographies of entries with an asterisk (*). Other abbreviations and symbols: *Rep.* = 'data representation,' *dep* = 'syntactic dependencies,' *phrase struc* = 'phrase structure,' *morpho* = 'morphological features,' *mod.* = 'modified,' *AAVE* = 'African-American Vernacular English,' *GML* = 'Middle Low Saxon,' *NLD* = 'Dutch,' *FRA* = 'French,' 🔒 = access is not immediate, 📍 = fine-grained dialect distinctions, ✎ = phonetic/phonemic transcription, ✎ = pronunciation spelling, A = LRL orthography, ¶ = normalized orthography.

Corpus	Langs	Annotation	Size	Rep.
TaPaCo (subset) (Scherrer, 2020) zenodo.org/record/3707949	NDS, GOS	paraphrases	1107 sents (NDS), 122 sents (GOS)	✎
Wenkersätze (Wenker, 1889–1923; Schmidt et al., 2020–) github.com/engsterhold/wenker-storage	DEU* 📍	translations (across dialects, DEU)	2210 samples×40 sents	✎/✎
SB-CH (subset) (Grubenmann et al., 2018) github.com/spinningbytes/SB-CH	GSW	sentiment	2.8k sents	✎
SwissDial (Dogan-Schönberger et al., 2021) projects.mtc.ethz.ch/swiss-voice-data-collection 🔒	GSW 📍, WAE	topic, translations (across dialects and into DEU)	2.5–4.6 hrs×8 lects	🎙✎¶
xSID/SID4LR (subset) (van der Goot et al., 2021; Aepli et al., 2023) bitbucket.org/robvanderg/sid4lr	GSW, BAR (South Tyrol)	slot and intent detection, translations (14 langs)	800 sents	✎

Table 2: **Corpora with semantic annotations or parallel sentences.** Abbreviations and symbols: *Rep.* = 'data representation,' 🔒 = access is not immediate, 📍 = fine-grained dialect distinctions, 🎙 = audio, ✎ = pronunciation spelling, ¶ = standard orthography. *The Wenkersätze contain samples from various German dialects, but those are not annotated directly (only the town names are shared).

5 Corpora

Most of the language varieties we survey have no or only a very recent written tradition. This unique challenge is reflected in the different written formats used to represent the data (if the corpora contain any written material at all) and concerns both the transcription of audio data (Tagliamonte, 2007; Gaeta et al., 2022) as well as the elicitation of written data (Millour and Fort, 2020). We opted to discern between audio data 🎙 and the following written variants: standard orthographies (of the doculects themselves where existing **A** (e.g., West Frisian orthography), or of a closely related higher-resource language otherwise ¶), ad-hoc pronunciation spelling (by speakers of the doculect) ✎, and phonetic or phonemic transcriptions (by linguists) ✎. Appendix B provides examples.

The following corpora are sorted by annotation and curation type. For an overview sorted by language, see Appendix A. Some of the corpora share the same data sources. Appendix C lists the cases where we are aware of such overlaps.

5.1 Annotated corpora

This section only includes corpora with manual (or manually corrected) annotations.

5.1.1 Morphosyntactic annotation

Table 1 provides an overview of datasets with morphosyntactic annotations. These mostly contain part-of-speech (POS) tags and/or syntactic dependencies. Such annotations are, for instance, of interest to dialectologists studying morphosyntactic variation (see for example Lybaert et al., 2019). Automatically generating high-quality morphosyntactic annotations for non-standard and/or low-resource data is not trivial, and the more annotated data are available for training, the better the results tend to be (Schulz and Ketschik, 2019; Scherrer et al., 2019a).

The annotation standards tend to either be general and cross-linguistically applicable (inviting comparisons between languages), or to be very specific to the language variety at hand. In the former case, annotations follow the guidelines from the Universal Dependencies project (Zeman et al., 2022) (UD, UPOS). In the latter case, tag sets created for a (usually closely related) higher-resource language are modified so that they capture the lower-resource language variety's idiosyncrasies. These specialized tag sets are based on: the annotations of the Giellatekno project (Wiechetek et al., 2022), the annotations developed for the Penn Parsed Corpora of Historical English (Penn-h),[11] the tag set of the Norwegian Dependency Treebank (NDT) (Solberg et al., 2014) (based on the Oslo-Bergen Tagger's tag set, OBT, (Johannessen et al., 2012)), and the Stuttgart-Tübingen tag set (STTS) (Schiller et al., 1999).

Most of the annotated corpora are presented only in one written form, typically either written in a standard orthography or pronunciation spelling.

contain more personally identifying information (like the voice of someone from a small speaker population), and it requires more work to censor locations or personal names in audio data than in text data (see also Seyfeddinipur et al., 2019).

[11] ling.upenn.edu/hist-corpora/annotation/index.html

Corpus	Langs	Size	Rep.
Føroyskur talumálsbanki (Jacobsen, 2022) clarino.uib.no/corpuscle-classic/corpus-list 🔒	FAO	599.9k toks	A
BLARK 1.0 (Background text corpus) (Simonsen et al., 2022) (incl. FTS (Språkbanken and Fróðskaparsetur Føroya) and Faroese Korp (Giellatekno)) maltokni.fo/en/resources	FAO	25M toks	A
Nordic Dialect Corpus (subset) (Johannessen et al., 2009) tekstlab.uio.no/nota/scandiasyn	NOR 📍, OVD 📍	1.9M toks (NOR), 15.7k toks (OVD)	¶ (NOR: ✎) (OVD: A)
LIA Norsk (Øvrelid et al., 2018) tekstlab.uio.no/LIA/korpus.html	NOR 📍	3.5M toks	✎ ¶ partially 🎙
Talemålsundersøkelsen i Oslo (TAUS) (Tekstlab, 2020) tekstlab.uio.no/nota/taus/	NOR (East/West Oslo) 📍	388k toks	✎ ¶
NorDial (Barnes et al., 2021) (subset) github.com/jerbarnes/nordial	NOR	348 tweets	✐
American Nordic Speech Corpus (CANS) (Johannessen, 2015) tekstlab.uio.no/norskiamerika/korpus.html	NOR (US/Canada) 📍, SWE (US) 📍	773k toks (NOR), 46k toks (SWE)	✎ ¶
Parallel dialectal–standard Swedish data (Hämäläinen et al., 2020; Ivars and Södergård, 2007) zenodo.org/record/4060296	SWE (Finland) 📍,	86.5k tokens	✎ ¶
Danish Gigaword (subset) (Strømberg-Derczynski et al., 2021; Kjeldsen, 2019) gigaword.dk	DAN (Bornholm)	ca. 400k tokens	unk.
Scottish Corpus of Texts & Speech (SCOTS) (subset) (Anderson et al., 2007) scottishcorpus.ac.uk	SCO	(unknown how many of 4.6M toks in SCO)	mix of ✐ ¶
Low Saxon Dialect Classification (LSDC) (Siewert et al., 2020) github.com/Helsinki-NLP/LSDC/	NDS, WEP, FRS, TWD, ACT 📍	88.9k sents	✐
LuxId (Lavergne et al., 2014) lrec2014.lrec-conf.org/en/shared-lrs/current-list-shared-lrs	LTZ/DEU/FRA code-switching	924 sents (most with LTZ content)	A
DiDi (subset) (Frey et al., 2019) hdl.handle.net/20.500.12124/7	BAR (South Tyrol)	unknown	✐
What's up, Switzerland? (Stark et al., 2014–2020; Ueberwasser and Stark, 2017) whatsup.linguistik.uzh.ch 🔒	GSW 📍	507k msgs / 3.6M toks	✐
Swatchgroup Geschäftsbericht (subset) (Graën et al., 2019) pub.cl.uzh.ch/wiki/public/pacoco/start	GSW	79.6k toks	✐
Text+Berg (subset) (Bubenhofer et al., 2015; Graën et al., 2019) textberg.ch/site/en/corpora 🔒 pub.cl.uzh.ch/wiki/public/pacoco/start	GSW	156 sents / 3.1k toks	✐
ArchiWals / CLiMAlp (Angster et al., 2017; Gaeta, 2020) climalp.org 🔒	WAE 📍	80+k tokens	✐

Table 3: **Other curated text corpora.** Abbreviations and symbols: *Rep.* = 'data representation,' 🔒 = access is not immediate, 📍 = fine-grained dialect distinctions, ✎ = phonetic/phonemic transcription, ✐ = pronunciation spelling, A = LRL orthography, ¶ = normalized orthography.

Corpus	Langs	Size	Rep.
BLARK 1.0 (Transcr. recordings) (Simonsen et al., 2022) maltokni.fo/en/resources	FAO 📍	100 h	🎙 **A** (some ✎)
Faroese Danish Corpus Hamburg (FADAC Hamburg) (subset) (Debess, 2019) corpora.uni-hamburg.de/hzsk/de/islandora/object/spoken-corpus:fadac-0.2.dan	FAO 📍	31 h	🎙 **A**
NB Tale – Speech Database for Norwegian (Språkbanken) nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-31/	NOR 📍	365 × 2 min (spon.), 7.6k sents (reading)	🎙 ✎ ¶
Norwegian Parliamentary Speech Corpus (NPSC) (Solberg and Ortiz, 2022) nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-58/	NOR 📍	140 h	🎙 ¶
Diachronic Electronic Corpus of Tyneside English (DECTE) (Corrigan et al., 2012) research.ncl.ac.uk/decte/index.htm 🔒	ENG (UK: Tyneside)	72 h / 804k toks	🎙 ¶ (some ✎)
Intonational Variation in English (IViE) (Nolan and Post, 2014) phon.ox.ac.uk/files/apps/IViE/	ENG (UK, Ireland) 📍	36 h	🎙 ¶
Crowdsourced high-quality UK and Ireland English Dialect speech data set (Demirsahin et al., 2020) openslr.org/83	ENG (UK, Ireland) 📍	31 h	🎙 ¶
Helsinki Corpus of British English Dialects (University of Helsinki, 2006) varieng.helsinki.fi/CoRD/corpora/Dialects/ 🔒	ENG (UK) 📍	1 M toks	🎙 ¶
Nationwide Speech Project (NSP) (Clopper and Pisoni, 2006) u.osu.edu/nspcorpus	ENG (US) 📍	60 × 1 hr	🎙 (some ¶)
Corpus of Regional African American Language (CORAAL) (Kendall and Farrington, 2021) oraal.uoregon.edu/coraal	ENG (AAVE) 📍	135.6 hrs / 1.5M toks	🎙 ¶
Common Voice Corpus 12.0 (subset) (Ardila et al., 2020) commonvoice.mozilla.org/en/datasets	FRY	150 h	🎙 **A**
Frisian AudioMining Enterprise (FAME) (Yılmaz et al., 2016) ru.nl/clst/tools-demos/datasets/ 🔒	FRY (some 📍)	18.5 h	🎙 **A**
Recordings of Dutch-Frisian council meetings (Bentum et al., 2022) frisian.eu/dutchfrisiancouncilmeetings	FRY	26 h / 281k toks	🎙 **A**
Corpus Spoken Frisian (Frisian Academy) www1.fa.knaw.nl/ksf.html 🔒	FRY	200 h (65 h transcribed)	🎙 (**A**)
Sprachvariation in Norddeutschland (SiN, Hamburg collection) (Schröder, 2011; Elmentaler et al., 2015) hdl.handle.net/11022/0000-0000-7EE3-3 🔒	NDS, FRS, DEU	300 h	🎙
Regional Variants of German 1 (RVG1) (Burger and Schiel, 1998) hdl.handle.net/11022/1009-0000-0004-3FF4-3	DEU* 📍	500+ × 1 min	🎙 ✎ ¶
Zwirner-Korpus (downloadable subset) (Zwirner and Bethge, 1958; IDS) dgd.ids-mannheim.de 🔒	NDS, WEP, SXU, VMF, BAR, GSW, DEU** 📍	3 h / 24.8k toks	🎙 ¶
Texas German Sample Corpus (TGSC) (Blevins, 2022) doi.org/10.18738/T8/IOX9ZA	DEU (Texas)	13.5 h / 75k toks	🎙 ¶
Audioatlas Siebenbürgisch-Sächsischer Dialekte (University of Munich) hdl.handle.net/11022/1009-0000-0001-27B9-3 🔒	DEU (Trans. Saxon)***	360 h / 450k toks	🎙 ¶ (some ✎)
CABank Yiddish Corpus (Newman, 2015) ca.talkbank.org/access/Yiddish.html	YID (New York)	1 hr	🎙 ✎
SXUCorpus (Herms et al., 2016) Contact authors 🔒	SXU 📍	500 min / 70k toks	🎙 ¶
Kontatti (Ghilardi, 2019) kontatti.projects.unibz.it 🔒	BAR (S. Tyrol), CIM	unknown	🎙 ¶
ArchiMob (Scherrer et al., 2019b) spur.uzh.ch/en/departments/research/textgroup/ArchiMob.html (audio files: 🔒)	GSW 📍	70 h	🎙 ✎ ¶
SDS-200 (Plüss et al., 2022) swissnlp.org/datasets/ 🔒	GSW	200 h	🎙 ¶
Swiss Parliaments Corpus (Plüss et al., 2021a) www.cs.technik.fhnw.ch/i4ds-datasets	GSW	293 h	🎙 ¶
Gemeinderat Zürich Audio Corpus (Plüss et al., 2021b) www.cs.technik.fhnw.ch/i4ds-datasets	GSW	1208 h	🎙
All Swiss German Dialects Test Set (Plüss et al., 2021b) www.cs.technik.fhnw.ch/i4ds-datasets	GSW, WAE 📍	13 h / 5.8k utterances	🎙 ¶
Walliserdeutsch/RRO (Garner, 2014; Garner et al., 2014) zenodo.org/record/4580286 🔒	WAE	8.3 h	🎙 ✐

Table 4: **Other audio corpora.** Abbreviations and symbols: *Rep.* = 'data representation,' 🔒 = access is not immediate, 📍 = fine-grained dialect distinctions, 🎙 = audio, ✎ = phonetic/phonemic transcription, ✐ = pronunciation spelling, **A** = LRL orthography, ¶ = normalized orthography. *It is unclear whether the RVG1 recordings are in regionally accented (Standard) German or whether they are in Low Saxon, Bavarian and other regional languages spoken in Germany, Switzerland, Austria and Northern Italy. **The Zwirner-Korpus contains samples from various dialects spoken in what used to be West Germany. ***Transylvanian Saxon is a variety of Moselle Franconian that does not have its own ISO code. It is more closely related to Luxembourgish than to Standard German.

Corpus	Languages and sizes
Tatoeba (subset; with >100 sents) `tatoeba.org/en/downloads`	in sentences: NDS (18.1k), YDD (12.8k), GOS (5.7k), FRR (2.9k), SWG (1.9k), LTZ (884), FRY (641), GSW (474), FAO (417), BAR (227)
Ubuntu `opus.nlpl.eu/Ubuntu.php`	in toks: NDS (35.3k), FRY (22.4k), FAO (20.2k), LIM (18.4k), LTZ (17.0k)
KDE4 `opus.nlpl.eu/KDE4-v2.php`	NDS (1.1M toks), FRY (0.3M toks), LTZ (28.8k toks)
GNOME `opus.nlpl.eu/GNOME.php`	NDS (0.7M toks), LIM (0.4M toks), FRY (55.7k toks)
Mozilla-I10n `mozilla-l10n/mt-training-data`	FRY (0.4M toks), LTZ (6.9k toks)
QED (Abdelali et al., 2014) `opus.nlpl.eu/QED.php`	LTZ (19.2k toks), FAO (6.4k toks)
TED2020 (Reimers and Gurevych, 2020) `opus.nlpl.eu/TED2020.php`	LTZ (1.7k toks)
Danish Gigaword (subset) (Strømberg-Derczynski et al., 2021) `gigaword.dk`	DAN (South Jutish) (ca. 20k tokens)
SwissCrawl (Linder et al., 2020) `icosys.ch/swisscrawl` 🔒	GSW (500k+ sents)
SB-CH (Grubenmann et al., 2018) `github.com/spinningbytes/SB-CH` 🔒	GSW (ca. 116k sents)
SwigSpot (Linder, 2018) `github.com/derlin/SwigSpot_Schwyzertuutsch-Spotting`	GSW (8k sents)
Web to Corpus (W2C) (subset) (Majliš, 2011; Majliš and Žabokrtský, 2012) `hdl.handle.net/11858/00-097C-0000-0022-6133-9`	in MB: YID (125), FAO (102), LTZ (81), FRY (72), SCO (35), NDS (24), LI (20)
CC-100 (subset) (Wenzek et al., 2020) `data.statmt.org/cc-100/`	FRY (174 MB), YID (51 MB), LIM (8.3 MB)
OSCAR (subset) (Abadji et al., 2022) `oscar-project.github.io/documentation/` 🔒	in toks: YID (14.3M), FRY (9.8M), LTZ (2.5M), NDS (1.6M), GSW (34k)
Wikipedia (subset) `dumps.wikimedia.org`	discussed in detail in Appendix D

Table 5: **Uncurated corpora.** 🔒 = Access not immediate. The corpora in the top section contain parallel sentences with many translations and are (also) distributed via the OPUS project (Tiedemann, 2012).

Some cases (marked with an asterisk* in the table) require further explanation: The Norwegian LIA and NDC treebanks (Øvrelid et al., 2018; Kåsen et al., 2022) use normalized orthographies (Nynorsk and Bokmål, respectively), but aligned versions of the original phonetic and orthographic transcriptions can be downloaded from the Tekstlab links in the table. The sentences in the UD Low Saxon LSDC treebank (Siewert et al., 2021) are presented both in the original ad-hoc pronunciation spelling and in a recently proposed orthography for Low Saxon, *Nysassiske Skryvwyse*. The Yiddish corpus (Santorini, 2021) is romanized, partially according to the YIVO transliteration system, and partially in a non-systematic manner.

5.1.2 Semantic annotation and parallel sentences

Very few resources with other types of annotations exist; we were able to find only five (Table 2), all of which have very different kinds of annotations: sentiment or topic classification, intent detection and slot-filling, translations and paraphrases.

5.1.3 Dialect annotation

Many corpora contain detailed annotations on the dialect area (or more precise location) an utterance's speaker or the author of a document is from. Such information is important for linguistic research comparing related dialects (Wieling and Nerbonne, 2015), for comparing the results of traditional and quantitative dialectological approaches (e.g. Heeringa et al., 2009) and for evaluating whether NLP systems perform differently on different closely related language varieties (Ziems et al., 2022). Since corpora with such annotations belong to all of the categories of curated datasets in this survey, they are not presented on their own, but instead marked with a pin symbol 📍 elsewhere.

5.2 Other curated corpora

5.2.1 Text corpora

Table 3 presents unannotated written corpora of low-resource languages like Elfdalian or Faroese, and corpora that showcase dialectal variation through phonetic transcriptions or pronunciation spelling. (While variation also occurs on linguistic levels encoded in normalized text written in stan-

dard orthographies – lexical, syntactic or pragmatic variation – we focus on phonological variation, as this is where specialized corpora are required.)

5.2.2 Audio corpora

In this survey, our focus lies on written resources, and as such, this selection of audio corpora is not exhaustive.[12] However, many of the language varieties surveyed in this article are predominately spoken rather than written. Creating language technology for unwritten languages is a topic of interest for NLP researchers (Scharenborg et al., 2020), and this is also reflected by the number of recently created speech corpora for Germanic LRLs.

Many of the audio corpora (Table 4) fall into one of two categories: recordings created for dialectological research, and post-hoc collections of already existing audio data (like radio broadcasts or public recordings of council meetings). Most of the audio corpora are at least partially transcribed, typically according to a standard orthography.

5.3 Uncurated text corpora

A final type of corpus are uncurated text collections (Table 5). This includes data coming from community-based data collection efforts unrelated to research projects (Wikipedia, Tatoeba) and open-source translations of (mostly) user interfaces, as well as web-crawled data.

It is important to note that there are quality issues with web-crawled corpora, especially for low-resource languages (Kreutzer et al., 2022).[13] Both CC-100 (Wenzek et al., 2020) and OSCAR (Abadji et al., 2022) are cleaned versions of CommonCrawl[14]– and Abadji et al. (2022) specifically remark on the low quality of the low-resource language data in that dataset.

Some of the translated corpora also have quality issues: the Low Saxon Ubuntu and GNOME corpora (Tiedemann, 2012) both also contain some Standard German content. We exclude subcorpora that contain mostly foreign language or non-linguistic material (for instance, the West Flemish QED subcorpus (Abdelali et al., 2014; Tiedemann, 2012)).

Wikipedia has editions in many Germanic low-resource languages and at different activity and contributor levels, as we survey in Appendix D. Projects extend wiki dumps with automatically inferred annotations (Pan et al., 2017; Schwenk et al., 2021), or release automatically aligned German–Alemannic/Bavarian bitext (Artemova and Plank, 2023).[15] The linguistic quality of LRL wikis is not always very high – the Scots Wikipedia made the news in 2020, when attention was brought to the fact that half of that wiki's articles had been created/edited by a non-Scots speaker writing in a parody of Scots (Brooks and Hern, 2020). Quality issues should be taken into account when working with data from small wikis without much oversight, e.g., with data or tools based on the Scots Wikipedia before clean-up started in fall 2020.[16]

6 Outlook

Creating NLP resources and technology for LRLs is an active field. At the time of writing this paper, several additional resources were concurrently under construction or revision: *UD Frisian Frysk*, a treebank for West Frisian (Heeringa et al., 2021),[17] *Boarnsterhim Corpus*, a West Frisian audio corpus (Sloos et al., 2018),[18] *Schweizerdeutsches Mundartkorpus*, a Swiss German text corpus (Weibel and Peter, 2020),[19] and the *Corpus of Southern Dutch Dialects* (Breitbarth et al., 2018).[20] Community-based projects are also being actively developed: many of the small Wikipedias have active editors (Appendix D), as do many of the Tatoeba collections. We welcome contributions to our companion website to track such progress.

Speaker populations of LRLs are not a monolith. Accordingly, different speaker communities have different interests in terms when it comes to the development of language technologies (Lent et al., 2022). The creation of downstream technologies made for public use should be made in accordance of the wishes and needs of the relevant speaker communities (see also Bird, 2022).

[12] Additional corpora documenting variation in spoken English can be found via the SPADE project (Stuart-Smith et al., 2017-2020).

[13] However, see Artetxe et al. (2022) for an argument that the linguistic quality of a corpus might not be the most important factor for all downstream applications.

[14] commoncrawl.org

[15] github.com/mainlp/dialect-BLI

[16] E.g., Scots is included in the language list of mBERT (Devlin et al., 2019), which was trained on Wikipedia data in 2019: github.com/google-research/bert/blob/master/multilingual.md

[17] github.com/UniversalDependencies/UD_Frisian-Frysk

[18] taalmaterialen.ivdnt.org/download/tstc-boarnsterhimcorpus1-0

[19] chmk.ch/de/info_all

[20] gcnd.ugent.be

We make the following **recommendations** for researchers who *work* with LRL datasets:

- Investigate the quality of uncurated data, as it might be especially poor for LRLs.
- Check whether (pre-)training, development and test data are truly from independent datasets – the dearth of high-quality LRL data means that datasets may be likely to overlap.
- Consider quantitative work by dialectologists and sociolinguists who might not publish in typical NLP venues.

To researchers who *create* such datasets, we recommend to:

- Document the transcription principles (if the data were originally in an audio format) / if any standardized orthographies were used (if the language variety does not have an official orthography).
- The low number of available high-quality datasets per language variety means that the impact of losing such a resource is much greater. Therefore, please upload your corpus to an archive geared towards long-term data storage (like the CLARIN Language Resource Inventory,[21] the LRE Map or Zenodo).
- Provide easy-to-find documentation with details on the corpus size, data sources and the annotation procedure.

7 Conclusion

We have presented an analysis of over 80 corpora containing data in Germanic low-resource languages, with a focus on non-standardized or only recently standardized varieties. We additionally share the corpus overview on a public companion website (`github.com/mainlp/germanic-lrl-corpora`) that can easily be updated as more language resources are released.

Acknowledgements

We thank the anonymous reviewers as well as the members of the MaiNLP research lab for their constructive feedback. This research is supported by European Research Council (ERC) Consolidator Grant DIALECT 101043235. This work was partially funded by the ERC under the European Union's Horizon 2020 research and innovation program (grant 740516).

[21] `clarin.eu/content/language-resource-inventory`

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a cleaner document-oriented multilingual crawled corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4344–4355, Marseille, France. European Language Resources Association.

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The AMARA corpus: Building parallel language resources for the educational domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1856–1862, Reykjavik, Iceland. European Language Resources Association (ELRA).

Noëmi Aepli and Simon Clematide. 2018. Parsing approaches for Swiss German. In *Proceedings of the 3rd Swiss Text Analytics Conference (SwissText), Winterthur, Switzerland*.

Noëmi Aepli, Çağrı Çöltekin, Rob van der Goot, Tommi Jauhiainen, Mourhaf Kazzaz, Nikola Ljubešić, Kai North, Barbara Plank, Yves Scherrer, and Marcos Zampieri. 2023. Findings of the VarDial evaluation campaign 2023. In *Proceedings of the Tenth Workshop on NLP for Similar Languages, Varieties and Dialects*, Dubrovnik, Croatia. Association for Computational Linguistics. To appear.

Alham Fikri Aji, Genta Indra Winata, Fajri Koto, Samuel Cahyawijaya, Ade Romadhony, Rahmad Mahendra, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, Timothy Baldwin, Jey Han Lau, and Sebastian Ruder. 2022. One country, 700+ languages: NLP challenges for underrepresented languages and dialects in Indonesia. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7226–7249, Dublin, Ireland. Association for Computational Linguistics.

Jean Anderson, Dave Beavan, and Christian Kay. 2007. SCOTS: Scottish corpus of texts and speech. *Creating and Digitizing Language Corpora: Volume 1: Synchronic Databases*, 1: Synchronic Databases:17–34.

Marco Angster, Marco Bellante, Raffaele Cioffi, and Livio Gaeta. 2017. I progetti DiWaC e ArchiWals. *Bollettino dell'Atlante Linguistico Italiano*, 41:83–94.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben

Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common Voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Ekaterina Artemova and Barbara Plank. 2023. Low-resource bilingual dialect lexicon induction with large language models. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*.

Mikel Artetxe, Itziar Aldabe, Rodrigo Agerri, Olatz Perez-de Viñaspre, and Aitor Soroa. 2022. Does corpus quality really matter for low-resource languages? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7383–7390. Association for Computational Linguistics.

Jennifer Bahr-Lamberti. 2016. Ressourcen zu deutschen Dialekten im Internet. *Zeitschrift für germanistische Linguistik*, 44(2):316–322.

Jeremy Barnes, Petter Mæhlum, and Samia Touileb. 2021. NorDial: A preliminary corpus of written Norwegian dialect use. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 445–451, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Martijn Bentum, Louis ten Bosch, Henk van den Heuvel, Simone Wills, Domenique van der Niet, Jelske Dijkstra, and Hans Van de Velde. 2022. A speech recognizer for Frisian/Dutch council meetings. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1009–1015, Marseille, France. European Language Resources Association.

Delphine Bernhard, Pascale Erhart, Dominique Huck, and Lucie Steiblé. 2019. Annotated corpus for the Alsatian dialects. Zenodo. Version 2.0.

Delphine Bernhard, Anne-Laure Ligozat, Fanny Martin, Myriam Bras, Pierre Magistry, Marianne Vergez-Couret, Lucie Steible, Pascale Erhart, Nabil Hathout, Dominique Huck, Christophe Rey, Philippe Reynés, Sophie Rosset, Jean Sibille, and Thomas Lavergne. 2018. Corpora with part-of-speech annotations for three regional languages of France: Alsatian, Occitan and Picard. In *11th edition of the Language Resources and Evaluation Conference*, Miyazaki, Japan.

Steven Bird. 2020. Decolonising speech and language technology. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3504–3519, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Steven Bird. 2022. Local languages, third spaces, and other high-resource scenarios. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7817–7829, Dublin, Ireland. Association for Computational Linguistics.

Margaret Blevins. 2022. Texas German sample corpus. Texas Data Repository.

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, Austin, Texas. Association for Computational Linguistics.

Anouck Braggaar and Rob van der Goot. 2021. Challenges in annotating and parsing spoken, code-switched, Frisian-Dutch data. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 50–58, Kyiv, Ukraine. Association for Computational Linguistics.

Anne Breitbarth, Melissa Farasyn, Anne-Sophie Ghyselen, and Jacques Van Keymeulen. 2018. Het Gesproken Corpus van de zuidelijk-Nederlandse Dialecten. *Handelingen – Koninklijke Zuid-Nederlandse maatschappij voor taal- en letterkunde en geschiedenis*, 72:23–38.

Libby Brooks and Alex Hern. 2020. Shock an aw: US teenager wrote huge slice of Scots Wikipedia. The Guardian.

Noah Bubenhofer, Martin Volk, Fabienne Leuenberger, and Daniel Wüest. 2015. Text+Berg–Korpus (release 151 v01). Digital edition of the SAC yearbook 1864-1923, Echo des Alpes 1872-1924, Die Alpen, Les Alpes, Le Alpi 1925-2014, The Alpine Journal 1969-2008. Institut für Computerlinguistik, Universität Zürich.

Susanne Burger and Florian Schiel. 1998. RVG 1 – A database for regional variants of contemporary German. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, pages 1083–1087, Granada, Spain.

Nicoletta Calzolari, Riccardo Del Gratta, Gil Francopoulo, Joseph Mariani, Francesco Rubino, Irene Russo, and Claudia Soria. 2012. The LRE map. harmonising community descriptions of resources. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1084–1089, Istanbul, Turkey. European Language Resources Association (ELRA).

Cynthia G. Clopper and David B. Pisoni. 2006. The Nationwide Speech Project: A new corpus of American English dialects. *Speech Communication*, 48.

Karen P. Corrigan, Isabelle Buchstaller, Adam Mearns, and Hermann Moisl. 2012. The diachronic electronic corpus of Tyneside English. Newcastle University.

Silvia Dal Negro and Simone Ciccolone. 2020. KONTATTO: A laboratory for the study of language contact in South Tyrol. *Sociolinguistica*, 34(1):241–247.

Iben Nyholm Debess. 2019. FADAC Hamburg 1.0. guide to the Faroese Danish corpus Hamburg. *Kieler Arbeiten zur skandinavistischen Linguistik*, 6.

Isin Demirsahin, Oddur Kjartansson, Alexander Gutkin, and Clara Rivera. 2020. Open-source multispeaker corpora of the English accents in the British Isles. In *Proceedings of The 12th Language Resources and Evaluation Conference (LREC)*, pages 6532–6541, Marseille, France. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Nina Dobrushina and Elena Sokur. 2022. Spoken corpora of Slavic languages. *Russian Linguistics*, 46:77–93.

Pelin Dogan-Schönberger, Julian Mäder, and Thomas Hofmann. 2021. SwissDial: Parallel multidialectal corpus of spoken Swiss German. *Computing Research Repository*, arXiv:2103.11401.

Michael Elmentaler, Joachim Gessinger, Jens Lanwer, Peter Rosenberg, Ingrid Schröder, and Jan Wirrer. 2015. Sprachvariation in Norddeutschland (SiN). In Roland Kehrein, Alfred Lameli, and Stefan Rabanus, editors, *Regionale Variation des Deutschen*, pages 397–424. De Gruyter.

Hanna Fischer and Juliane Limper. 2019. Regionalsprachliche Forschungsergebnisse online. In Joachim Herrgen and Jürgen Erich Schmidt, editors, *Deutsch*, Sprache und Raum – Ein internationales Handbuch der Sprachvariation, pages 879–897. De Gruyter Mouton, Berlin, Boston.

Jennifer-Carmen Frey, Aivars Glaznieks, and Egon W. Stemle. 2019. DIDI - the DiDi corpus of South Tyrolean CMC 1.0.0. Eurac Research CLARIN Centre.

Frisian Academy. Corpus spoken Frisian. Department of Social Sciences (Frisian Academy) and Department of Linguistics (Frisian Academy).

Livio Gaeta. 2020. The Observer's Paradox meets corpus linguistics: Written and oral sources for the Walser linguistic islands in Italy. In *Endangered linguistic varieties and minorities in Italy and the Balkans*, Vienna. VLACH.

Livio Gaeta, Marco Angster, Raffaele Cioffi, and Marco Bellante. 2022. Corpus linguistics for low-density varieties. minority languages and corpus-based morphological investigations. *Corpus*, 23.

Philip Garner. 2014. Walliserdeutsch. Zenodo.

Philip N. Garner, David Imseng, and Thomas Meyer. 2014. Automatic speech recognition and translation of a Swiss German dialect: Walliserdeutsch. In *Proceedings of Interspeech*, pages 2118–2122, Singapore.

Marta Ghilardi. 2019. Eliciting comparable spoken data in minor languages: first observations from the corpus Kontatti. *Suvremena lingvistika*, 45(88):231–246.

Giellatekno. KORP version 6.0.1, Faroese texts.

Rob van der Goot, Ibrahim Sharaf, Aizhan Imankulova, Ahmet Üstün, Marija Stepanović, Alan Ramponi, Siti Oryza Khairunnisa, Mamoru Komachi, and Barbara Plank. 2021. From masked language modeling to translation: Non-English auxiliary tasks improve zero-shot spoken language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2479–2497, Online. Association for Computational Linguistics.

Johannes Graën, Tannon Kew, Anastassia Shaitarova, and Martin Volk. 2019. Modelling large parallel corpora: The Zurich Parallel Corpus Collection. In *Proceedings of the 7th Workshop on Challenges in the Management of Large Corpora (CMLC)*, pages 1–8. Leibniz-Institut für Deutsche Sprache.

Ralf Grubenmann, Don Tuggener, Pius von Däniken, Jan Deriu, and Mark Cieliebak. 2018. SB-CH: A Swiss German corpus with sentiment annotations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Imane Guellil, Houda Saâdane, Faical Azouaou, Billel Gueni, and Damien Nouvel. 2021. Arabic natural language processing: An overview. *Journal of King Saud University – Computer and Information Sciences*, 33(5):497–507.

Mika Hämäläinen, Niko Partanen, and Khalid Alnajjar. 2020. Normalization of different Swedish dialects spoken in Finland. In *Proceedings of the 4th ACM SIGSPATIAL Workshop on Geospatial Humanities*, page 24–27, New York, NY, USA. Association for Computing Machinery.

Harald Hammarström, Robert Forkel, Martin Haspelmath, and Sebastian Bank. 2022. Glottolog 4.7. Leipzig: Max Planck Institute for Evolutionary Anthropology. Available online at http://glottolog.org, accessed on 2023-02-03.

Wilbert Heeringa, Gosse Bouma, Martha Hofman, Eduard Drenth, Jan Wijffels, and Hans Van de Velde. 2021. POS tagging, lemmatization and dependency parsing of West Frisian.

Wilbert Heeringa, Keith Johnson, and Charlotte Gooskens. 2009. Measuring Norwegian dialect distances using acoustic features. *Speech Communication*, 51(2):167–183.

Robert Herms, Laura Seelig, Stefanie Münch, and Maximilian Eibl. 2016. A corpus of read and spontaneous Upper Saxon German speech for ASR evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4648–4651, Portorož, Slovenia. European Language Resources Association (ELRA).

Nora Hollenstein and Noëmi Aepli. 2015. A resource for natural language processing of Swiss German dialects. In *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology, GSCL 2015*, pages 108–109.

Dirk Hovy and Christoph Purschke. 2018. Capturing regional variation with distributed place representations and geographic retrofitting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4383–4394, Brussels, Belgium. Association for Computational Linguistics.

IDS. Datenbank für gesprochenes Deutsch (DGD), Deutsche Mundarten: Zwirner-Korpus.

Anton Karl Ingason, Eiríkur Rögnvaldsson, Einar Freyr Sigurðsson, and Joel C. Wallenberg. 2012. Faroese parsed historical corpus (FarPaHC) 0.1. CLARIN-IS.

Ann-Marie Ivars and Lisa Södergård. 2007. Spara det finlandssvenska talet. In *Nordisk dialektologi og sociolingvistik: Foredrag på 8. Nordiske Dialektologkonference Århus 15.–18. august 2006*, pages 202–206. Aarhus Universitet.

Jógvan í Lon Jacobsen. 2022. Flertalsformer af ari-ord i den færøske talesprogsbank. *Nordlyd*, 46(1):103–113.

Janne Bondi Johannessen. 2015. The corpus of American Norwegian speech (CANS). In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 297–300, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Janne Bondi Johannessen, Kristin Hagen, André Lynum, and Anders Nøklestad. 2012. OBT+ stat. a combined rule-based and statistical tagger. In Gisle Andersen, editor, *Exploring Newspaper Language: Using the web to create and investigate a large corpus of modern Norwegian*, volume 49 of *Studies in Corpus Linguistics*, page 51. John Benjamins Publishing.

Janne Bondi Johannessen, Joel James Priestley, Kristin Hagen, Tor Anders Åfarli, and Øystein Alexander Vangsnes. 2009. The Nordic Dialect Corpus–an advanced research tool. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, pages 73–80, Odense, Denmark. Northern European Association for Language Technology (NEALT).

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Andre Kåsen, Kristin Hagen, Anders Nøklestad, Joel Priestly, Per Erik Solberg, and Dag Trygve Truslew Haug. 2022. The Norwegian Dialect Corpus treebank. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4827–4832, Marseille, France. European Language Resources Association.

Tyler Kendall and Charlie Farrington. 2021. The corpus of regional African American language. Version 2021.07. Eugene, OR: The Online Resources for African American Language Project.

Alex Speed Kjeldsen. 2019. Bornholmsk ordbog, version 2.0. *Maal og Maele*, 40(2):22–31.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Harm Lameris and Sara Stymne. 2021. Whit's the richt pairt o speech: PoS tagging for Scots. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 39–48, Kiyv, Ukraine. Association for Computational Linguistics.

Thomas Lavergne, Gilles Adda, Martine Adda-Decker, and Lori Lamel. 2014. Automatic language identity

tagging on word and sentence-level in multilingual text sources: A case-study on Luxembourgish. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3300–3304, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jérémy Leixa, Valérie Mapelli, and Khalid Choukri. 2014. Inventaire des ressources linguistiques des langues de France. Version 1.1. Evaluations and Language resources Distribution Agency (ELDA).

Heather Lent, Kelechi Ogueji, Miryam de Lhoneux, Orevaoghene Ahia, and Anders Søgaard. 2022. What a creole wants, what a creole needs. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6439–6449, Marseille, France. European Language Resources Association.

Lucy Linder. 2018. SwigSpot – creation of a Swiss German dataset. Master's thesis, University of Applied Sciences and Arts Western Switzerland.

Lucy Linder, Michael Jungo, Jean Hennebert, Claudiu Cristian Musat, and Andreas Fischer. 2020. Automatic creation of text corpora for low-resource languages from the internet: The case of swiss german. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2706–2711, Marseille, France. European Language Resources Association.

Chloé Lybaert, Bernard De Clerck, Jorien Saelens, and Ludovic De Cuypere. 2019. A corpus-based analysis of V2 variation in West Flemish and French Flemish dialects. *Journal of Germanic Linguistics*, 31(1):43–100.

Petter Mæhlum, Andre Kåsen, Samia Touileb, and Jeremy Barnes. 2022. Annotating Norwegian language varieties on Twitter for part-of-speech. In *Proceedings of the Ninth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 64–69, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Manuel Mager, Ximena Gutierrez-Vasques, Gerardo Sierra, and Ivan Meza-Ruiz. 2018. Challenges of language technologies for the indigenous languages of the Americas. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 55–69, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Martin Majliš. 2011. W2C – Web to Corpus – corpora. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Martin Majliš and Zdeněk Žabokrtský. 2012. Language richness of the web. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2927–2934, Istanbul, Turkey. European Language Resources Association (ELRA).

Alice Millour and Karën Fort. 2018. À l'écoute des locuteurs : production participative de ressources langagières pour des langues non standardisées. *Revue TAL*.

Alice Millour and Karën Fort. 2020. Text corpora and the challenge of newly written languages. In *1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020)*, Proceedings of the 1st Joint SLTU and CCURL Workshop, Marseille, France.

Stephen Morey, Mark W. Post, and Victor A Friedman. 2013. The language codes of ISO 639: A premature, ultimately unobtainable, and possibly damaging standardization. Talk given at the PARDISEC RRR Conference, December 2013.

John Nerbonne, Wilbert Heeringa, Jelena Prokić, and Martijn Wieling. 2021. Dialectology for computational linguists. In Marcos Zampieri and Preslav Nakov, editors, *Similar Languages, Varieties, and Dialects: A Computational Perspective*, Studies in Natural Language Processing, pages 96–118. Cambridge University Press.

Zelda Kahan Newman. 2015. Discourse markers in the narratives of New York Hasidim. more V2 attrition. In Janne Bondi Johannessen and Joseph C. Salmons, editors, *Germanic heritage languages in North America. Acquisition, attrition and change*, pages 178–197. John Benjamins.

Francis Nolan and Brechtje Post. 2014. The IViE Corpus. In *The Oxford Handbook of Corpus Phonology*. Oxford University Press.

Sebastian Nordhoff and Harald Hammarström. 2011. Glottolog/Langdoc: Defining dialects, languages, and language families as collections of resources. In *Proceedings of the First International Workshop on Linked Science*.

Lilja Øvrelid, Andre Kåsen, Kristin Hagen, Anders Nøklestad, Per Erik Solberg, and Janne Bondi Johannessen. 2018. The LIA treebank of spoken Norwegian dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Jean-Marie Pierrel, Christophe Parisse, Jérôme Blanchard, Etienne Petitjean, and Frédéric Pierre. 2017.

ORTOLANG: a French infrastructure for Open Resources and TOols for LANGuage. In *Selected papers from the CLARIN Annual Conference 2016, Aix-en-Provence, 26–28 October 2016, CLARIN Common Language Resources and Technology Infrastructure*, volume 136, pages 102–112. Linköping University Electronic Press.

Michel Plüss, Manuela Hürlimann, Marc Cuny, Alla Stöckli, Nikolaos Kapotis, Julia Hartmann, Malgorzata Anna Ulasik, Christian Scheller, Yanick Schraner, Amit Jain, Jan Deriu, Mark Cieliebak, and Manfred Vogel. 2022. SDS-200: A Swiss German speech to standard German text corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3250–3256, Marseille, France. European Language Resources Association.

Michel Plüss, Lukas Neukom, Christian Scheller, and Manfred Vogel. 2021a. Swiss Parliaments Corpus, an automatically aligned Swiss German speech to Standard German text corpus. In *Proceedings of the Swiss Text Analytics Conference 2021*.

Michel Plüss, Lukas Neukom, and Manfred Vogel. 2021b. SwissText 2021 task 3: Swiss German speech to Standard German text. In *Proceedings of the Swiss Text Analytics Conference 2021*.

Alan Ramponi. 2022. NLP for language varieties of Italy: Challenges and the path forward. *Computing Research Repository*, arXiv:2209.09757.

Georg Rehm, Maria Berger, Ela Elsholz, Stefanie Hegele, Florian Kintzel, Katrin Marheinecke, Stelios Piperidis, Miltos Deligiannis, Dimitris Galanis, Katerina Gkirtzou, Penny Labropoulou, Kalina Bontcheva, David Jones, Ian Roberts, Jan Hajič, Jana Hamrlová, Lukáš Kačena, Khalid Choukri, Victoria Arranz, Andrejs Vasiljevs, Orians Anvari, Andis Lagzdiņš, Jūlija Meļņika, Gerhard Backfried, Erinç Dikici, Miroslav Janosik, Katja Prinz, Christoph Prinz, Severin Stampler, Dorothea Thomas-Aniola, José Manuel Gómez-Pérez, Andres Garcia Silva, Christian Berrío, Ulrich Germann, Steve Renals, and Ondrej Klejch. 2020. European Language Grid: An overview. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3366–3380, Marseille, France. European Language Resources Association.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Eiríkur Rögnvaldsson, Anton Karl Ingason, Einar Freyr Sigurðsson, and Joel Wallenberg. 2012. The Icelandic parsed historical corpus (IcePaHC). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1977–1984, Istanbul, Turkey. European Language Resources Association (ELRA).

Beatrice Santorini. 2021. Penn parsed corpus of historical Yiddish. Version 1.0.

Odette Scharenborg, Laurent Besacier, Alan Black, Mark Hasegawa-Johnson, Florian Metze, Graham Neubig, Sebastian Stüker, Pierre Godard, Markus Müller, Lucas Ondel, Shruti Palaskar, Philip Arthur, Francesco Ciannella, Mingxing Du, Elin Larsen, Danny Merkx, Rachid Riad, Liming Wang, and Emmanuel Dupoux. 2020. Speech technology for unwritten languages. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:964–975.

Yves Scherrer. 2020. TaPaCo: A corpus of sentential paraphrases for 73 languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6868–6873, Marseille, France. European Language Resources Association.

Yves Scherrer, Tanja Samardžić, and Elvira Glaser. 2019a. Digitising Swiss German: how to process and study a polycentric spoken language. *Language Resources and Evaluation*, 53:837–863.

Yves Scherrer, Tanja Samardžić, and Elvira Glaser. 2019b. ArchiMob: Ein multidialektales Korpus schweizerdeutscher Spontansprache. *Linguistik Online*, 98(5):425–454.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS (kleines und großes Tagset).

Jürgen Erich Schmidt, Joachim Herrgen, Roland Kehrein, Alfred Lameli, and Hanna Fischer. 2020–. Regionalsprache.de (REDE III. Forschungsplattform zu den modernen Regionalsprachen des Deutschen. Edited by Robert Engsterhold, Heiko Girnth, Simon Kasper, Juliane Limper, Georg Oberdorfer, Tillmann Pistor, Anna Wolańska. Assisted by Dennis Beitel, Milena Gropp, Maria Luisa Krapp, Vanessa Lang, Salome Lipfert, Jeffrey Pheiff, Bernd Vielsmeier.

Manuela Schönenberger and Eric Haeberli. 2019. Ein geparstes und grammatisch annotiertes Korpus schweizerdeutscher Spontansprachdaten. In *Germanistische Linguistik*, volume 241–243, pages 79–104.

Ingrid Schröder. 2011. Sprachvariation in Norddeutschland (SiN). Archived in Hamburger Zentrum für Sprachkorpora. Version 0.1.

Sarah Schulz and Nora Ketschik. 2019. From 0 to 10 million annotated words: part-of-speech tagging for Middle High German. *Language Resources and Evaluation*, 53:837–863.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main*

Volume, pages 1351–1361, Online. Association for Computational Linguistics.

Mandana Seyfeddinipur, Felix Ameka, Lissant Bolton, Jonathan Blumtritt, Brian Carpenter, Hilaria Cruz, Sebastian Drude, Patience L Epps, Vera Ferreira, Ana Vilacy Galucio, Brigit Hellwig, Oliver Hinte, Gary Holton, Dagmar Jung, Irmgarda Kasinskaite Buddeberg, Manfred Krifka, Susan Kung, Miyuki Monroig, Ayu'Nwi Ngwabe Neba, Sebastian Nordhoff, Brigitte Pakendorf, Kilu von Prince, Felix Rau, Keren Rice, Michael Riessler, Vera Szoelloesi Brenig, Nick Thieberger, Paul Trilsbeek, Hein van Der Voort, and Tony Woodbury. 2019. Public access to research data in language documentation: Challenges and possible strategies. *Language Documentation & Conservation*, 13:545–563.

Abdulhadi Shoufan and Sumaya Alameri. 2015. Natural language processing for dialectical Arabic: A survey. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 36–48, Beijing, China. Association for Computational Linguistics.

Janine Siewert, Yves Scherrer, and Jörg Tiedemann. 2021. Towards a balanced annotated Low Saxon dataset for diachronic investigation of dialectal variation. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*, pages 242–246, Düsseldorf, Germany. KONVENS 2021 Organizers.

Janine Siewert, Yves Scherrer, Martijn Wieling, and Jörg Tiedemann. 2020. LSDC – a comprehensive dataset for Low Saxon dialect classification. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 25–35, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

Gary Simons and Steven Bird. 2003. The Open Language Archives Community: An Infrastructure for Distributed Archiving of Language Resources. *Literary and Linguistic Computing*, 18(2):117–128.

Annika Simonsen, Sandra Saxov Lamhauge, Iben Nyholm Debess, and Peter Juel Henrichsen. 2022. Creating a basic language resource kit for Faroese. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4637–4643, Marseille, France. European Language Resources Association.

Marjoleine Sloos, Eduard Drenth, and Wilbert Heeringa. 2018. The Boarnsterhim corpus: A bilingual Frisian-Dutch panel and trend study. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Per Erik Solberg and Pablo Ortiz. 2022. The Norwegian parliamentary speech corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1003–1008, Marseille, France. European Language Resources Association.

Per Erik Solberg, Arne Skjærholt, Lilja Øvrelid, Kristin Hagen, and Janne Bondi Johannessen. 2014. The Norwegian dependency treebank. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 789–795, Reykjavik, Iceland. European Language Resources Association (ELRA).

Språkbanken. NB Tale – speech database for Norwegian. National Library of Norway. Last updated 2015-12-22.

Språkbanken and Fróðskaparsetur Føroya. FTS – Faroese text collection. Språkbanken Text (Department of Swedish at the University of Gothenburg) and the University of Faroe Islands. Last modified: 2015-05-27.

Elisabeth Stark, Simone Ueberwasser, and Anne Göhring. 2014–2020. Corpus "What's up, Switzerland?". University of Zurich.

STIH. 2020. Bisame_gsw (alsacien) : corpus brut et annoté. ORTOLANG (Open Resources and TOols for LANGuage).

Leon Strømberg-Derczynski, Manuel Ciosici, Rebekah Baglini, Morten H. Christiansen, Jacob Aarup Dalsgaard, Riccardo Fusaroli, Peter Juel Henrichsen, Rasmus Hvingelby, Andreas Kirkedal, Alex Speed Kjeldsen, Claus Ladefoged, Finn Årup Nielsen, Jens Madsen, Malte Lau Petersen, Jonathan Hvithamar Rystrøm, and Daniel Varab. 2021. The Danish Gigaword corpus. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 413–421, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Jane Stuart-Smith, Morgan Sonderegger, and Jeff Mielke. 2017-2020. SPeech Across Dialects of English (SPADE): Large-scale digital analysis of a spoken language across space and time.

Sali A. Tagliamonte. 2007. Representing real language: Consistency, trade-offs and thinking ahead! In Joan C. Beal, Karen P. Corrigan, and Hermann L. Moisl, editors, *Creating and Digitizing Language Corpora*, volume 1: Synchronic Databases, pages 205–240. Palgrave Macmillan UK, London.

Tekstlab. 2020. TAUS – the spoken language investigation in Oslo. Version 3.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Francis Tyers, Mariya Sheyanova, Aleksandra Martynova, Pavel Stepachev, and Konstantin Vinogorodskiy. 2018. Multi-source synthetic treebank creation

for improved cross-lingual dependency parsing. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 144–150, Brussels, Belgium. Association for Computational Linguistics.

Simone Ueberwasser and Elisabeth Stark. 2017. What's up, Switzerland? a corpus-based research project in a multilingual country. *Linguistik Online*, 84(5).

University of Helsinki. 2006. The Helsinki corpus of British English dialects. Department of Modern Languages, University of Helsinki.

University of Munich. Audioatlas siebenbürgisch-sächsischer Dialekte (ASD). Institut für deutsche Kultur und Geschichte Südosteuropas, Institut für romanische Philologie, IT-Gruppe Geisteswissenschaften. LMU Munich.

Jacques Van Keymeulen, Veronique De Tier, Anne Breitbarth, Anne-Sophie Ghyselen, and Melissa Farasyn. 2019. Het dialectologische corpus 'Stemmen uit het verleden' van de Universiteit Gent. *Volkskunde*, 120(2):193–204.

Dieter Van Uytvanck, Claus Zinn, Daan Broeder, Peter Wittenburg, and Mariano Gardellini. 2010. Virtual Language Observatory: The portal to the language resources and technology universe. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Elaine Vaughan and Brian Clancy. 2016. Sociolinguistic information and Irish English corpora. In Raymond Hickey, editor, *Sociolinguistics in Ireland*, pages 365–388. Palgrave Macmillan, London.

Wietse de Vries, Martijn Wieling, and Malvina Nissim. 2022. Make the best of cross-lingual transfer: Evidence from POS tagging with over 100 languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7676–7685, Dublin, Ireland. Association for Computational Linguistics.

Manuela Weibel and Muriel Peter. 2020. Compiling a large Swiss German dialect corpus. In *Proceedings of the 5th Swiss Text Analytics Conference (SwissText) & 16th Conference on Natural Language Processing (KONVENS)*.

Georg Wenker. 1889–1923. *Sprachatlas des Deutschen Reichs*. Marbug. Handdrawn by Emil Maurmann, Georg Wenker and Ferdinand Wrede. Published online as 'Digitaler Wenker-Atlas'.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Linda Wiechetek, Katri Hiovain-Asikainen, Inga Lill Sigga Mikkelsen, Sjur Moshagen, Flammie Pirinen, Trond Trosterud, and Børre Gaup. 2022. Unmasking the myth of effortless big data - making an open source multi-lingual infrastructure and building language resources from scratch. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1167–1177, Marseille, France. European Language Resources Association.

Martijn Wieling and John Nerbonne. 2015. Advances in dialectometry. *Annual Review of Linguistics*, 1(1):243–264.

Emre Yılmaz, Maaike Andringa, Sigrid Kingma, Jelske Dijkstra, Frits van der Kuip, Hans Van de Velde, Frederik Kampstra, Jouke Algra, Henk van den Heuvel, and David van Leeuwen. 2016. A longitudinal bilingual Frisian-Dutch radio broadcast database designed for code-switching research. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4666–4669, Portorož, Slovenia. European Language Resources Association (ELRA).

Jihene Younes, Emna Souissi, Hadhemi Achour, and Ahmed Ferchichi. 2020. Language resources for Maghrebi Arabic dialects' NLP: A survey. *Language Resources and Evaluation*, 54(4):1079–1142.

Marcos Zampieri, Preslav Nakov, and Yves Scherrer. 2020. Natural language processing for similar languages, varieties, and dialects: A survey. *Natural Language Engineering*, 26(6):595–612.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Salih Furkan Akkurt, Gabrielė Aleksandravičiūtė, Ika Alfina, Avner Algom, Chiara Alzetta, Erik Andersen, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Glyd Aranes, Maria Jesus Aranzabe, Bilge Nas Arıcan, Þórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Katla Ásgeirsdóttir, Deniz Baran Aslan, Cengiz Asmazoğlu, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Juan Belieni, Kepa Bengoetxea, Yifat Ben Moshe, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, Kristina Brokaitė, Aljoscha Burchardt,

Marie Candito, Bernard Caron, Gauthier Caron, Lauren Cassidy, Maria Clara Castro, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Liyanage Chamila, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Daniela Corbetta, Marine Courtin, Mihaela Cristescu, Philemon Daniel, Elizabeth Davidson, Leonel Figueiredo de Alencar, Mathieu Dehouck, Martina de Laurentiis, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Christian Ebert, Hanne Eckhoff, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Federica Favero, Jannatul Ferdaousi, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Federica Gamba, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Takahiro Harada, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Marivel Huerta Mendez, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Artan Islamaj, Kaoru Ito, Siratun Jannat, Tomáš Jelínek, Apoorva Jha, Katharine Jiang, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Ritván Karahóǧa, Boris Katz, Tolga Kayadelen, Sarveswaran Kengatharaiyer, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Köse, Alexey Koshevoy, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sandra Kübler, Adrian Kuqi, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yixuan Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Krister Lindén, Nikola Ljubešić, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Ilya Makarchuk, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Stella Markantonatou, Héctor Martínez Alonso, Lorena Martín Rodríguez, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Tatiana Merzhevich, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Hulda Óladóttir, Adédayọ̀ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Noam Ordan, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Teresa Paccosi, Alessio Palmero Aprosio, Anastasia Panova, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Giulia Pedonese, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Andrea Peverelli, Jason Phelan, Jussi Piitulainen, Rodrigo Pintucci, Tommi A Pirinen, Emily Pitler, Magdalena Plamada, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Robert Pugh, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Mizanur Rahoman, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Mathilde Regnault, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eiríkur Rögnvaldsson, Ivan Roksandic, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Ben Rozonoyer, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Marta Sartor, Mitsuya Sasaki, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang

Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Maria Shvedova, Janine Siewert, Einar Freyr Sigurðsson, João Ricardo Silva, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Haukur Barri Símonarson, Kiril Simov, Dmitri Sitchinava, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Barbara Sonnenhauser, Shafi Sourov, Carolyn Spadine, Rachele Sprugnoli, Vivian Stamou, Steinþór Steingrímsson, Antonio Stella, Abishek Stephen, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Daniel Swanson, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Mirko Tavoni, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Sara Tonelli, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sveinbjörn Þórðarson, Vilhjálmur Þorsteinsson, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Elena Vagnoni, Sowmya Vajjala, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Uliana Vedenina, Giulia Venturi, Eric Villemonte de la Clergerie, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Shira Wigderson, Sri Hartati Wijono, Vanessa Berwanger Wille, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Anna Zhuravleva, and Rayan Ziane. 2022. Universal Dependencies 2.11. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Caleb Ziems, William Held, Jingfeng Yang, and Diyi Yang. 2022. Multi-VALUE: A framework for cross-dialectal English NLP. *Computing Research Repository*, arXiv:2212.08011.

Eberhard Zwirner and Wolfgang Bethge. 1958. *Erläuterungen zu den Texten*. Number 1 in Lautbibliothek der deutschen Mundarten. Vandenhoek & Ruprecht.

A Resources by language

We include the languages associated with the ISO 639-3 codes FAO (Faroese), OVD (Elfdalian), SCO (Scots), FRR (North Frisian), FRY (West Frisian), STQ (Saterland Frisian), NDS (Low Saxon), FRS (East Frisian Low Saxon), GOS (Gronings), TWD (Twents), ACT (Achterhoeks), WEP (Westphalian), ZEA (Zeelandic), VLS (West Flemish), LTZ (Luxembourgish), LIM (Limburgish), KSH (Colognian), PFL (Palatine German), PDC (Pennsylvania Dutch), YID (Yiddish), SXU (Upper Saxon), VMF (East Franconian), BAR (Bavarian), SWG (Swabian), GSW (Swiss German and Alsatian), WAE (Walser), and CIM (Cimbrian). Our survey also encompasses data for dialects/non-standard varieties of Norwegian (NOR), Swedish (SWE), Danish (DAN), English (ENG), and German (DEU) that do not have their own ISO codes.

We use ISO codes to refer to (groups of) language varieties for practical reasons – despite their shortcomings as labels for varieties from linguistic continua (Morey et al., 2013; Nordhoff and Hammarström, 2011), they are widely used and recognized, and many of the corpora in this survey are described in terms that easily map to ISO codes.

In some cases, the codes or the corpus descriptions are ambiguous. For instance, many Low Saxon corpora contain entries that also belong to one of the more specific Dutch Low Saxon codes, and some Swiss German corpora also contain some Walser content. Where possible (and where the data instances themselves are labelled on a precise enough level), we use the more specific codes.

Table 6 provides an overview of resource types by language variety.

B Written representations

Table 7 provides examples of different types of written representations and showcases how diverse each category can be.

Examples 1a, 2a, 3a, 4a/b, 5a and 6a are written in **standardized orthographies** (or in lower-cased versions of standard orthographies with no pronunciation). Of these, sentences 1a, 4a and 5a are written in orthographies developed for their respective low-resource languages **A**, while 2a, 3a, 4b and 6a are normalized and written in the orthographies of closely related standard languages ¶ (the last two are Elfdalian written in Swedish and Swiss German written in Standard German).

Sentences 5b and 7a present two examples of ad-hoc **pronunciation spellings** ✎. These kinds of spellings vary from speaker to speaker, and one and the same speaker might also choose different spellings of the same word at different times.

Phonetic or phonemic transcriptions ✎ have

Language		Dialect/ Location	Morpho-syntax	Semantic	Parallel (curated)	Uncurated text	Curated data
North Germanic							
FAO	Faroese	📍	✔			✔	🎙 ✎ **A**
NOR	(non-std.) Norwegian	📍	✔				🎙 ✎ ✎ ¶
OVD	Elfdalian	📍					**A** ¶
SWE	(non-std.) Swedish	📍					✎ ¶
DAN	(non-std.) Danish	📍				✔	?
Anglo-Frisian							
SCO	Scots		✔			✔	✎ **A** ¶
ENG	(non-std.) English	📍	✔				🎙 ¶
FRY	West Frisian	📍	✔			✔	🎙 **A**
FRR	North Frisian					✔	
STQ	Saterland Frisian					✔	
*Low German**							
NDS	Low Saxon	📍	✔		✔	✔	🎙 ✎ **A**
FRS	East Frisian Low Saxon					✔	🎙
GOS	Gronings				✔	✔	
TWD	Twents					✔	✎
ACT	Achterhoeks					✔	✎
WEP	Westphalian						🎙 ¶
Macro-Dutch							
VLS	West Flemish	📍	✔			✔	✎
ZEA	Zeelandic					✔	
Middle German							
LTZ	Luxembourgish					✔	**A**
KSH	Colognian					✔	
LIM	Limburgish					✔	
PFL	Palatine German					✔	
PDC	Pennsylvania Dutch					✔	
YID	Yiddish**		✔			✔	🎙 ✎
SXU	Upper Saxon						🎙 ¶
Upper German							
DEU	(non-std.) German				✔		🎙 ✎ ¶
VMF	East Franconian						🎙 ¶
BAR	Bavarian		✔	✔	✔	✔	🎙 ✎ ✎ ¶
CIM	Cimbrian						🎙 ¶
SWG	Swabian					✔	
GSW	Swiss Ger. & Alsatian	📍	✔	✔	✔	✔	🎙 ✎ ✎ ¶
WAE	Walser	📍		✔	✔	✔	🎙 ✎

Table 6: **Corpora by language variety.** For ease of reference, the language are sorted by Germanic subbranches (based on Glottolog (Hammarström et al., 2022)). *For additional texts written in varieties of Low German/Saxon with other ISO 639-3 codes, see the note on the Low Saxon Wikipedias in Table 8. **Glottolog discerns between Eastern Yiddish (Middle German) and Western Yiddish (Upper German). Symbols: 🎙 = audio, ✎ = phonetic/phonemic transcription, ✎ = pronunciation spelling, **A** = LRL orthography, ¶ = normalized orthography.

From the Faroese BLARK recordings (Simonsen et al., 2022):		
1a	**A**	vit høvdu matpakka við og eg hugnaði mær óført
1b	✎	vId h9dI m%EApaHga v%i: o e h%u:najI mar %OW:f9zd
1c	✎	vɪd hœdɪ ˈmɛa̯pʰaʰga ˈviː o e ˈhuːnajɪ maɪ ˈɔu̯ːfœs̬d
“We had lunchboxes with us and I enjoyed myself greatly.”		
From the Norwegian NB Tale corpus (Språkbanken):		
2a	¶	Etter litt godsnakk kom tre av kyrne mot han mens den fjerde glei og fall
2b	✎	""{t@4 l"it g""u:snAkk k"Om t4"e: "A:v C"y:n‘@ m"u:t "An m"ens d_= fj""{:d‘@ gl"eI "O: f"Al
2c	✎	²ɛtəɪ ¹lɪt ²guːsnɑkk ¹kɔm ¹tɪeː ¹ɑːv ¹çyːŋə ¹muːt ¹ɑn ¹mɛns dn̩ ²fjæːɖə ¹glɛɪ̯ ¹oː ¹fɑl
“After some coaxing, three of the cows came towards him while the fourth one slipped and fell.”		
From the Norwegian part of the Nordic Dialect Corpus (Johannessen et al., 2009):		
3a	¶	det er slik at de fleste kommer jo att når de får # unger
3b	✎	de e sjlik att dæi fLeste kjemme jo att nårr dæi fær # onnga
“The thing is that most people return when they have *[brief pause]* kids.”		
From the Elfdalian part of the Nordic Dialect Corpus (Johannessen et al., 2009):		
4a	**A**	wen wa wen war eð før ien månað ? juni ?
4b	¶	vad va- vad var det för en månad ? juni ?
“What, wha-, what month was it? June?”		
From UD Low Saxon LSDC (Siewert et al., 2021):		
5a	**A**	Nu leyt em de böyse vynd disse nacht kyn ouge an enander doon.
5b	✐	Nu leit em de baise Find düse Nacht kinn Auge an enander dohn.
“Now the wicked enemy didn’t let them get a wink of sleep that night.”		
From the Swiss German ArchiMob corpus (Scherrer et al., 2019b):		
6a	¶	können sie ihre jugendzeit beschreiben
6b	✎	chönd sii iri jugendziit beschriibe
“Can you describe your youth?”		
From the BISAME corpus (STIH, 2020):		
7a	✐	Niema hat salamols gweßt as die Werter vum franzeescha kumma.
“Nobody knew then that these words came from French.”		

Table 7: **Examples of written representations.** Symbols: ✎ = phonetic/phonemic transcription, ✐ = pronunciation spelling, **A** = LRL orthography, ¶ = normalized orthography.

different styles depending on each corpus's transcription guidelines. Examples 1b and 2b are written in modified versions of SAMPA and X-SAMPA, and the corpora come with sufficient documentation to automatically convert these transcriptions into IPA (1c, 2c). (The superscript symbols [1] and [2] in example 2c are commonly used to indicate the Norwegian pitch accent.) The transcription styles presented in examples 3b and 6b are based on Norwegian and Standard German orthography, respectively. What sets them apart from pronunciation spellings is that they are consistent across the entire corpus and that they follow linguistic rationales that often are outlined in the corpus documentation.

C Overlapping data sources

Several of the corpora mentioned in this article overlap with each other:

- *UD Faroese OFT* and the *Korp* subcorpus of the background corpus of the Faroese *BLARK 1.0* contain material from the Faroese Wikipedia.
- The *NDC Treebank* uses data from the *Nordic Dialect Corpus.*
- The *LIA Treebank* and *UD Norwegian NynorskLIA* are annotated subsets of *LIA Norsk*, and they overlap with each other.
- The *POS-tagged Scots corpus* contains annotated sentences from *SCOTS.*
- *UD Low Saxon LSDC* and *LSDC* overlap.
- *UD Frisian/Dutch Fame* is an annotated subset if *FAME.*
- Many of the sentences in *UD Swiss German UZH* are also in *NOAH's corpus.* Both of these corpora contain material from the Alemannic Wikipedia.
- *SB-CH* contains *NOAH's corpus.*
- The *Annotated Corpus for the Alsatian Dialects* contains articles from the Alemannic Wikipedia that were explicitly tagged as Alsatian.
- *TaPaCo* is a subset of *Tatoeba.*
- Any corpus that includes data from the internet might overlap with the uncurated datasets in Section 5.3.

D Wikipedia statistics

Table 8 provides a comparison of Wikipedia sizes and user (vs. bot) activity.[22] The sizes of the small Germanic Wikipedias vary considerably from wiki to wiki (there are just under 2k Pennsylvania Dutch articles, while the (German) Low Saxon Wikipedia has over 84k articles), as does the number of recently active contributors (from 6 active non-bot users per month for Ripurarian/Colognian, Palatine German and Pennsylvania Dutch to 70 for Scots).

While bots can be used for automating many tasks that are unrelated to the textual diversity of a wiki (e.g., cleaning up article redirection pages), they can also be used to automatically create short template-based articles.[23] The share of manual edits (i.e., edits not by bots) is very varied across wikis – only about a quarter of all edits in the Pennsylvania Dutch Wikipedia have been made manually, compared to 79 % in the North Frisian Wikipedia. However, there is a clear trend towards a much larger proportion of manual edits: the vast majority of edits made only in the past year were manual edits.

Some of the wikis are written according to one or more orthographies, while others either do not include any spelling recommendations at all or encourage editors to use whatever pronunciation spelling they prefer. The Dutch Low Saxon Wikipedia, for instance, recommends *Nysassiske Skryvwyse*, whereas the German Low Saxon Wikipedia recommends another orthography: *Sass-Schrievwies*. The Ripurarian/Colognian Wikipedia, conversely, encourages idiosyncratic spellings.[24]

[22] The data sources are the automatically updated list of Wikipedia sizes at `meta.wikimedia.org/wiki/List_of_Wikipedias_by_language_group#Germanic` (last accessed 2023-01-31) and Wikimedia's metadata via `wikimedia.org/api/rest_v1/`. The scripts are available via `github.com/mainlp/wikistats`.

[23] For an example for the latter, see `nds.wikipedia.org/wiki/Bruker:ArtikelBot`

[24] These are the pages detailing orthographic conventions we were able to find (sorted by wiki size): `nds.wikipedia.org/wiki/Wikipedia:Sass`; `sco.wikipedia.org/wiki/Wikipedia:Spellin_an_grammar`; `als.wikipedia.org/wiki/Hilfe:Schrybig`; `bar.wikipedia.org/wiki/Wikipedia:Wia_schreib_i_a_guads_Boarisch%3F`; `frr.wikipedia.org/wiki/Wikipedia:Språkekoordinasjoon`; `li.wikipedia.org/wiki/Wikipedia:Wie_sjrief_ich_Limburgs`; `vls.wikipedia.org/wiki/Wikipedia:Gebruuk_van_streektoaln`; `nds-nl.wikipedia.org/wiki/Wikipedia:Spelling`; `stq.wikipedia.org/wiki/`;

Several of these wikis include (some) articles with metadata specifying which variety the document is written in.[25]

Wikipedia:Hälpe_bie_ju_seelter_Sproake; ksh.wikipedia.org/wiki/Wikipedia:Schrievwies

[25] Sorted by wiki size: nds.wikipedia.org/wiki/Kategorie:Artikels_na_Dialekt; als.wikipedia.org/wiki/Kategorie:Wikipedia:Dialekt; bar.wikipedia.org/wiki/Kategorie:Artikel_nach_Dialekt; frr.wikipedia.org/wiki/Kategorie:Spriakwiisen; li.wikipedia.org/wiki/Categorie:Wikipedia:Artikele_nao_dialek; vls.wikipedia.org/wiki/Categorie:Wikipedia:Artikels_noar_dialect; nds-nl.wikipedia.org/wiki/Kategorie:Nedersaksies_artikel; ksh.wikipedia.org/wiki/Saachjrupp:Wikipedia:Atikkel_ier_Shprooche; pfl.wikipedia.org/wiki/Sachgrubb:Adiggel_noch_em_Dialegd

Wikipedia & Language		Articles (01/2023)	Manual edits (2001–2022)	Manual edits (2022)	Monthly editors (2022)
nds	NDS (Germany)* (📍)	84 k	44 %	99 %	30
lb	LTZ	61 k	43 %	85 %	56
fy	FRY	50 k	60 %	99 %	54
sco	SCO	39 k	53 %	63 %	70
als	GSW + SWG + WAE (📍)	30 k	69 %	100 %	58
bar	BAR (📍)	27 k	68 %	63 %	39
frr	FRR (📍)	17 k	79 %	85 %	16
yi	YID	15 k	49 %	97 %	35
li	LIM	14 k	42 %	75 %	21
fo	FAO	14 k	41 %	99 %	29
vls	VLS (📍)	8 k	45 %	79 %	16
nds-nl	NDS (Netherlands)* (📍)	8 k	40 %	68 %	14
zea	ZEA	6 k	47 %	98 %	10
stq	STQ	4 k	38 %	81 %	8
ksh	KSH + other Ripuarian (📍)	3 k	32 %	99 %	6
pfl	PFL + oth. Rhen. Franc., Hessian (📍)	3 k	65 %	72 %	6
pdc	PDC	2 k	27 %	92 %	6
en	ENG	6608 k	90 %	92 %	102 574
de	DEU	2765 k	91 %	93 %	16 141
nl	NLD	2114 k	68 %	66 %	3521
da	DAN	289 k	63 %	64 %	711
is	ISL	56 k	54 %	79 %	118

Table 8: **Wikipedia statistics.** 'Manual edits' include the proportion of edits (of content pages) performed by registered non-bot users or anonymous editors (out of the total number of content page edits performed by anyone, including bots). The number of monthly editors is the mean number of registered non-bot users who edited at least one content page, per month. English, German, Dutch (NLD), Danish (DAN) and Icelandic (ISL) are included for comparison. The wikis with a pin symbol 📍 contain (some) articles tagged by dialect; see footnote 25. *The *nds* and *nds-nl* wikis are primarily concerned with varieties of Low Saxon spoken in, respectively, Germany and the Netherlands. The former also contains articles written in varieties associated with the ISO 639-3 codes WEP and FRS, and the latter with ACT, FRS, GOS, DRT (Drents), SDZ (Sallands), STL (Stellingwerfs), TWD and VEL (Veluws).

You say tomato, I say the same: A large-scale study of linguistic accommodation in online communities

Aleksandrs Berdicevskis
Språkbanken Text
Department of Swedish, Multilingualism, Language Technology
Gothenburg University
aleksandrs.berdicevskis@gu.se

Viktor Erbro
Chalmers University of Technology
erbro@student.chalmers.se

Abstract

An important assumption in sociolinguistics and cognitive psychology is that human beings adjust their language use to their interlocutors. Put simply, the more often people talk (or write) to each other, the more similar their speech becomes. Such accommodation has often been observed in small-scale observational studies and experiments, but large-scale longitudinal studies that systematically test whether the accommodation occurs are scarce. We use data from a very large Swedish online discussion forum to show that linguistic production of the users who write in the same subforum does usually become more similar over time. Moreover, the results suggest that this trend tends to be stronger for those pairs of users who actively interact than for those pairs who do not interact. Our data thus support the accommodation hypothesis.

1 Introduction

Language is a tool not only for conveying information, but also for expressing attitudes, constructing identities and building relationships (Eckert, 2012). One manifestation of this fundamental property of language is that how we speak (or write) depends on whom we are speaking (or writing) to. How exactly the audience affects the linguistic production is a complex and multi-faceted process which can be approached from various perspectives. Consider, for instance, the audience design theory (Bell, 1984), social identity theory (Reid and Giles, 2008) and accommodation theory (Giles, 1973; Gallois et al., 1995).

In this paper, we perform a large-scale test of the hypothesis that people adjust their production style to their interlocutors. This phenomenon is known as *accommodation* (sometimes *attunement* or *linguistic alignment*) or *convergence* if the styles of the interlocutors are becoming more similar (*divergence* if they are becoming more different). While it has received considerable attention within sociolinguistics (Rickford et al., 1994; Cukor-Avila and Bailey, 2001) and cognitive psychology (Garrod et al., 2018), large-scale longitudinal studies are wanting. An exception is a study by Nardy et al. (2014), who have observed a group of French-speaking children at a kindergarten for one year and shown that children who interacted more frequently adopted similar usages of a number of sociolinguistic variables (such as, for instance, the dropping of the consonant /R/ in post-consonantal word-final positions).

Internet and social media in particular provide us with a vast amount of data about how people communicate and how they use language for other purposes than information transmission (Nguyen and P. Rosé, 2011). While in some respects these data are not as informative as those collected by direct observation or experimenting, in some other respects they may be equally or even more useful, providing very detailed information about who interacted when with whom and how. Besides, it is often possible to collect large datasets that enable more systematic hypothesis testing.

We use data from a very large Swedish discussion forum (Flashback) to test a widely held sociolinguistic assumption that "the more often people talk to each other, the more similar their speech will be" (Labov, 2001, p.288). In brief, we find pairs of Flashback users which during some period of time have actively interacted (see Section 2.2 for the definition of "active interaction"). We define a measure of linguistic distance between users and show that it is valid for our purposes (see Section 2.3). For every pair of users, we then calculate the linguistic distance between the two users' production before they have started interacting (Δ_{before}) and after it (Δ_{after}), and the difference between these distances ($\Delta_i = \Delta_{before} - \Delta_{after}$). If the convergence assumption is correct, we expect that the distance will tend to become smaller and the aver-

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 415–424

age Δ_i will be positive.

A positive Δ_i, however, can arise for different reasons, of which arguably the most prominent one is that distances between users become smaller not because users accommodate to specific interlocutors, but rather converge on a certain style adopted in the community (Danescu-Niculescu-Mizil et al., 2013). To test whether this is a better explanation, we perform a similar calculation for those pairs who have never had a single interaction, comparing texts written earlier (Δ_{early}) and later (Δ_{later}) during their activity on the forum ($\Delta_n = \Delta_{early} - \Delta_{later}$). If there is a convergence to norm, the average Δ_n should be positive.

It is also possible that both pairwise accommodation and convergence to the community norm occur simultaneously. Moreover, they might even be parts of the same process: if speakers do converge on a certain norm, this convergence can emerge (at least partly) due to pairwise interactions. It is, however, also possible that only one of these processes occurs. Speakers can, for instance, converge on the community norm by adjusting to some perceived "average" style and not specific individual interlocutors. On the other hand, it can be imagined that speakers do adjust to the individual interlocutors, but that does not lead to the emergence of the community norm (for instance, because different interlocutors are "pulling" in different directions). The purpose of this study is to provide some insight into these not entirely understood processes.

We envisage four likely outcomes of our experiments, summarized in Table 1. Other outcomes are possible, but would be more difficult to explain. We would, for instance, be surprised if Δ_n turns out to be larger than Δ_i (since if there is convergence to community norm, it should be affecting actively interacting and non-interacting users in approximately the same way). Another unexpected result would be a negative value of either Δ_n or Δ_i, since that would imply systematic divergence (see discussion in Section 4).

2 Materials and methods

2.1 Corpora

We use Flashback,[1] a very large Swedish discussion forum covering a broad variety of topics which has existed for more than two decades. In 2021, the proportion of internet users in Sweden (excluding those younger than eight years) who visited the forum at least once during the last 12 months was estimated to be 24% (Internetstiftelsen, 2021).

The forum is divided into 16 subforums, of which we use five in the main experiment: *Dator och IT* 'Computer and IT', *Droger* 'Drugs', *Hem, bostad och familj* 'Home, house and family', *Kultur & Media* 'Culture and media', *Sport och träning* 'Sport and training'. These five were selected as being relatively large, of comparable size and representing diverse and not directly related topics. In addition, we use a smaller subforum *Fordon och trafik* 'Vehicles and traffic' to evaluate our distance metric (see section 2.3).

To access the Flashback texts, we use the corpora created and maintained by Språkbanken Text, a Swedish national NLP infrastructure. The corpora are available for download[2] and for searching via the Korp interface (Borin et al., 2012) and its API.[3]

The basic corpus statistics are summarized in Table 2. The earliest available posts date back to 2000, and the corpora were last updated in February 2022. The number of users is estimated as a number of unique non-empty usernames. We list separately the number of "prolific" users, and we consider users prolific if they have written 6000 tokens or more. All other users will be discarded (many of the prolific users will not pass additional thresholds either, see Section 2.4).

Subforums may be further divided into subsub- and subsubsubforums, which we do not take into account. What is important for our purposes is that messages (posts) are always organized in threads: there is an initial message which starts a thread (often a question) and then an unlimited number of messages which either respond to the original message or to later messages or in some other way are related to the thread's topic. The structure of the thread is linear: that is, messages are posted in a strictly chronological order.

2.2 Defining interaction

Two users are assumed to have had an interaction if they have written messages within the same thread, the two messages are separated by no more than two other messages and there has gone no more than five days between the two messages were posted. This definition has been used by Hamilton et al. (2017) and Del Tredici and Fernández (2018),

[1] https://www.flashback.org/

[2] https://spraakbanken.gu.se/resurser?s=flashback&language=All

[3] https://ws.spraakbanken.gu.se/docs/korp

	Outcome	Interpretation
1	$\Delta_i > \Delta_n > 0$	Both pairwise accommodation and overall convergence to community norm are detected
2	$\Delta_i = \Delta_n > 0$	No pairwise accommodation; overall convergence to community norm is detected
3	$\Delta_i > \Delta_n = 0$	Pairwise accommodation is detected; no convergence to community norm
4	$\Delta_i = \Delta_n = 0$	No pairwise accommodation; no convergence to community norm

Table 1: Four likely outcomes of the experiment. Δ_i is the change of linguistic distance between actively interacting users, Δ_n is the change of distance between non-interacting users.

Subforum	tokens	users	prolific users
Computer	316M	187K	9.3K
Drugs	257M	123K	8.0K
Culture	434M	211K	12.2K
Home	348M	168K	10.0K
Sport	251M	105K	5.4K

Table 2: Basic statistics about the Flashback subforums. Prolific users have written 6000 tokens or more

but without the temporal threshold. We consider the temporal threshold useful, since Flashback can have very long threads, sometimes spanning over the years.

See the definition of "actively interacting users" in section 2.4.

2.3 Measuring linguistic distance

Potential solutions. A traditional sociolinguistic approach would be to identify a number of linguistic variables (features for which variation is known to exist) and use them for comparison (Nardy et al., 2014).The main problem with this approach is that most variables are not very frequent and it is thus difficult to collect enough observations. A traditional NLP approach would be to use a language model (Danescu-Niculescu-Mizil et al., 2013). Here, the main problem would be to ensure that the model has enough training data. We use a metric which is often applied in authorship attribution studies, Cosine Delta (Smith and Aldridge, 2011), a modification of Burrows' delta (Burrows, 2002). Its main advantage is that it can often be successfully applied to relatively small datasets, and it is also computationally efficient. It can also be considered a language model, though a very simple unigram-based one.

Cosine Delta. To calculate Cosine Delta between two texts, the texts are represented as t-dimensional vectors where every element is a z-score (standard score) of the relative frequency of one of t most frequent words. The cosine of the angle between the two vectors gauges their proximity, by subtracting it from 1, we get the distance (see Equation 1).

$$\Delta_\angle(T, T') = 1 - \frac{\mathbf{z}(T) \cdot \mathbf{z}(T')}{||\mathbf{z}(T)||_2 ||\mathbf{z}(T')||_2} \quad (1)$$

Cosine Delta has been shown to outperform Burrows' Delta and other similar measures (Jannidis et al., 2015; Evert et al., 2015).

Evaluating the metric. A typical usage of Cosine Delta is to compare text X of unknown or disputed authorship with texts by authors A and B in order to see whose style is more similar to the one used in X and whether the similarity is strong enough to attribute the text. This is not the same task that we have in mind. We want to compare texts written by authors A and B at time P and then at a later time Q in order to see whether the styles of the two authors have become more similar. In other words, we are not trying to infer who authored which text (we know that). Instead, we want to be able to measure the distance between two different authors.

To test whether Cosine Delta is suitable for that, we run the following experiment. The main requirement for an evaluation is a meaningful benchmark which can represent the ground truth. In order to evaluate a distance measure we need a set of texts between which true distances are known. We create such a set by mixing texts produced by two authors in different proportions. For two Flashback users (A0 and A1), an equal amount of tokens is extracted and used to create six texts: Base (contains solely the A0 production), 1 (80% of production belongs to A0, 20% to A1; every token is randomly selected), 2 (60% A0, 40% A1), 3 (40% A0, 60% A1), 4 (20% A0, 80% A1) and 5 (100% A1), see Figure 1.

We accept as ground truth that the distance between the Base text and, say, Text 1 should be

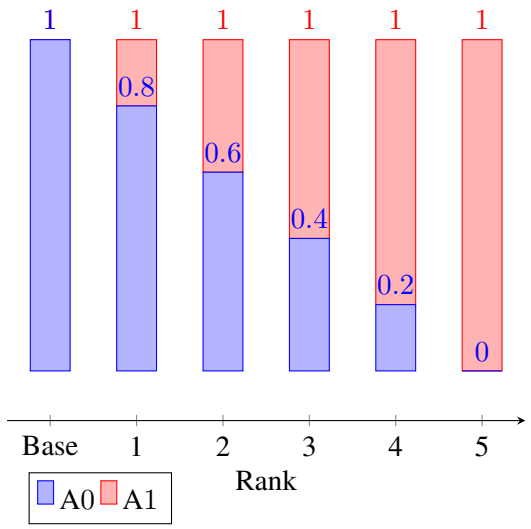


Figure 1: The artificial benchmark for evaluating the linguistic distance measure: six texts with different proportions of the authors' (A0 and A1) production.

n	t	ρ	Δ
1500	150	0.936 (0.1)	0.16 (0.06)
1500	300	0.936 (0.1)	0.15 (0.06)
1500	450	0.940 (0.1)	0.15 (0.06)
1500	600	0.944 (0.1)	0.15 (0.06)
3000	150	0.950 (0.1)	0.15 (0.07)
3000	300	0.952 (0.1)	0.14 (0.06)
3000	450	0.952 (0.1)	0.13 (0.06)
3000	600	0.952 (0.1)	0.13 (0.06)
4500	150	0.976 (0)	0.14 (0.08)
4500	300	0.978 (0)	0.13 (0.07)
4500	450	0.978 (0)	0.13 (0.07)
4500	600	0.978 (0)	0.13 (0.07)
6000	150	0.994 (0)	0.14 (0.06)
6000	300	0.994 (0)	0.13 (0.07)
6000	450	0.994 (0)	0.13 (0.07)
6000	600	0.994 (0)	0.13 (0.06)

Table 3: Evaluating Cosine Delta on 50 ground-truth sets. n is the number of tokens in the compared texts, t is the number of frequent words used to construct the vector, ρ is the average Spearman correlation coefficient, Δ is the average difference between authors A0 and A1 (between base and text 5). Interquartile ranges are provided in parentheses.

smaller than between Base and Text 5. We use Cosine Delta to compare Texts 1–5 with the Base text, rank them by their distance from Base and then measure Spearman correlation coefficient between this ranking and the true one (1, 2, 3, 4, 5).

We run the ranking test on 50 artificial sets, each consisting of six texts generated from two different authors' production, as described above. All data were extracted from the subforum *Fordon och trafik* 'Vehicles and traffic' (not used in the main experiment). The data were extracted consecutively without any randomization, i.e. the extraction script started from the beginning of the corpus, tried to extract a predefined number of tokens for every new user it encountered and stopped when it collected enough data for 100 unique users.

We try several combinations of two parameters: t, the dimension of vectors (the number of the most frequent words the frequencies of which will be used), and n, the minimum size of the texts to be compared (larger texts are expected yield more reliable estimates). The frequency list is compiled using the whole Flashback corpus (uncased). The results are reported in Table 3.

The performance of the ranking system is very high and increases as n increases. Unfortunately, increasing n decreases sample size, since less user pairs will be able to pass the thresholds (see Section 2.4). We judge that the best balance between reliability of Cosine Delta and sample size is reached with $n = 3000$ ($\rho \geq 0.95$). For $n = 6000$, the performance of Cosine Delta is better, but sample sizes (number of analyzable user pairs) are too small. We use $t = 300$, since larger values do not yield any gain for the chosen n values. Using Pearson correlation coefficient instead of Spearman yields approximately the same results (the values are 1-2 percentage points lower, but the trends are almost the same).

We also calculate average distance between authors A0 and A1 (that is, between Base and Text 5) to obtain a very rough estimate of average distance between two different users. Later, when we measure how linguistic distance changes over time, we will use this estimate as a reference point, something to compare the change against, so that we can judge how large the effect size is. For $n = 3000$ and $t = 300$, the average distance is about 0.13 (though there is, unsurprisingly, considerable variation).

Topic sensitivity. An important potential problem with measures like Cosine Delta is that they are topic-sensitive, that is, the distance values can be affected not only by differences in the authors' styles, but also by the topic, i.e., what the specific texts are about (Mikros and Argir, 2007; Björklund and Zechner, 2017). This is extremely undesirable for

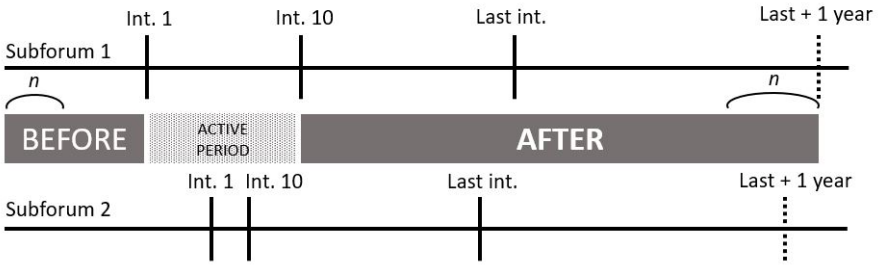


Figure 2: A visualization of how the periods **before** and **after** the active interaction has started are defined. Vertical lines represent interactions, the horizontal lines represent time. n earliest tokens are sampled from the "before" period, n latest tokens are sampled from "after"

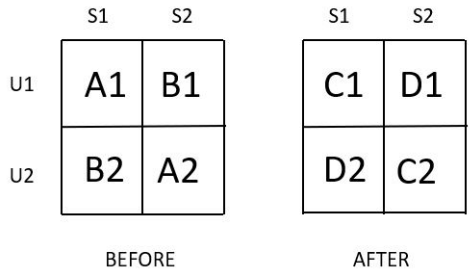


Figure 3: Visualization of the threshold requirements. Let the table cells represent how many tokens the **U**ser has written in the **S**ubforum in the given period. The following condition must be met for the user pair to be accepted: $((A1 \geq n\ AND\ A2 \geq n)\ OR\ (B1 \geq n\ AND\ B2 \geq n))\ AND\ ((C1 \geq n\ AND\ C2 \geq n)\ OR\ (D1 \geq n\ AND\ D2 \geq n))$

our purposes, since there is a risk that we observe that a convergence which is not in fact linguistic: the two authors do not start writing in a more similar way, they just start writing about more related topics. To eliminate or at least mitigate this risk, we always compare authors A and B by using texts that A wrote in one subforum and B in another subforum. While it is not completely impossible that the authors discuss similar topics in different subforums, it seems unlikely that "topical convergence" will systematically occur across subforums.

Note also that in the evaluation experiment described above all users come from the same subforum. Moreover, their production was extracted from the corpus consecutively and thus at least parts of it come from the same threads. That means that the users are likely to discuss related topics, and the ranking system must be able to capture differences in style despite potential similarities in topic, which it does very well.

2.4 Calculating distance change

As mentioned in Section 2.3, all our calculations are always based on two subforums at once (for instance, Home and Sport or Drugs and Computer). We will call such pairs of subforums *duplets* (to distinguish them from user *pairs*).

Two users are considered to have gone through a *period of active interaction* if they have had at least 10 interactions within a year in each of the subforums (that is, no less than 20 interactions in total). By requiring that the users actively interact in both subforums we ensure that there is a theoretical reason to expect convergence in both subforums and that the data are generally more comparable. We compare the production of users before and after the active interaction period, but ignore the period itself.

Within a subforum, the active period can have any length from one day to 365 days. We do not measure how often the users interact after the active period, but we discard all texts that have been produced more than one year later after the last interaction (it may be that users continue to interact and there are no messages to discard).

In other words, the general idea is that production before the active period includes everything written before the first interaction, production after the active period includes everything written after the tenth interaction (given that it is no more than one year apart from the first interaction), but no later than one year after the last interaction. We are, however, dealing with two subforums at once, and thus have two dates for each of the three seminal interactions. For convenience, we want the active period to be defined in the same way for both subforums. We achieve that by using the earlier of the dates for the first interaction and the later of the dates for the tenth generation (this can lead to

joint active period being longer than a year). When discarding the messages that were written after the users have stopped interacting (if any), we use the later of the last interaction dates. See the visual summary in Figure 2.

Users who have never had a single interaction are labelled as non-interacting. We compare them to actively interacting users and ignore all that end up in between: that is, have had some interactions but failed to pass the criteria outlined above (e.g. have had less than 10 interactions in total or have had more, but never 10 within a year). The reason for that is that we want the difference between groups (non-interacting and actively interacting users) to be as large as possible, so that potentially small effects can become visible.

Remember that we always want the linguistic distance to be calculated using text from different subforums. The procedure is as follows. For every pair, if before the active period, User 1 has produced at least n ($n = 3000$) tokens in Subforum 1, and User 2 has produced at least n tokens in Subforum 2, we calculate the distance between them, taking n tokens for User 1 from Subforum 1 and n tokens for User 2 from Subforum 2.

Obviously, if User 1 has n or more tokens in Subforum 2, and User 2 has n or more tokens in Subforum 1, the distance is calculated using tokens from Subforum 2 for User 1 and from Subforum 1 for User 2. If both conditions are met (Condition 1: User 1 has n or more tokens in Subforum 1 and User 2 has n or more tokens in Subforum 2; Condition 2: User 1 has n or more tokens in Subforum 2 and User 2 has n or more tokens in Subforum 1), we calculate both cross-subforum distances and use their arithmetic mean as the final result. If neither of the conditions is met, the pair is discarded. This procedure is visualized in Figure 3. The same user can occur unlimited times in different pairs.

Note that when we calculate distance between users A and B, we always use the same amount of tokens (n) for A and B (since using texts of different sizes might skew the Cosine Delta). For the "before" period, we extract the earliest n tokens, for the "after" period, the latest n ones (see Figure 2). The idea is to maximize the temporal distance between the periods in order to see stronger effect.

For non-interacting users, it is not obvious how to define "before" and "after", since the active period is not defined. We do the following: find the earliest first interaction date and the latest last interaction date across all actively interacting pairs. Then we take the date which is exactly in the middle between those two as the active period (the length of the active period is thus one day, which is common for interacting pairs, too). Then exactly the same procedure as for actively interacting pairs is applied, using the middle date to divide the data into "before" and "after".

There are many more non-interacting pairs than actively interacting ones, and calculating the distance change for all of them is computationally expensive. We go through the list of all non-interacting pairs in a randomized order and stop when m pairs have met the conditions, where m is five times the number of actively interacting pairs that have met the conditions. The reason for this decision is that the number of actively interacting pairs is rather small for some combinations of the subforums, and it makes sense to have somewhat larger samples at least for the non-interacting group.

3 Results

We perform the comparisons for all possible combinations of subforums (ten duplets in total). The results are summarized in Table 4. For every duplet and every type of user pair (actively interacting vs. non-interacting) we report sample size, average distance change ($\Delta_{before} - \Delta_{after}$) and the proportion of pairs for which the change was positive (the distance became smaller). Results for samples where the number of pairs is less than 20 are not reported.

Remember that in the evaluation experiment (Section 2.3) we roughly estimated the average distance between two different users to be around 0.13 for the chosen parameter values. While there clearly is large variation, and while the average distance can be larger for the main experiment (since the users' texts come from different subforums, not the same one), the estimate still provides us with a reference point and helps to put the observed distance changes in perspective. For Home-Sport-i, for instance, the average change is 0.033, which is approximately 25% of 0.13. This means that on average, actively interacting users in this duplet change their styles so much that they cover one quarter of an average distance the styles of two different persons.

Overall, the distance tends to become shorter both for interacting and non-interacting pairs. The

Subforum1	Subforum2	type	pairs	positive	change	IQR
home	sport	i	29	0.828	0.033	0.042
home	sport	n	145	0.524	-0.012	0.081
computer	drugs	i	15	-	-	-
computer	drugs	n	75	0.680	0.048	0.096
sport	drugs	i	67	0.612	0.015	0.110
sport	drugs	n	335	0.546	0.002	0.094
home	computer	i	46	0.630	0.060	0.121
home	computer	n	230	0.617	0.029	0.089
home	drugs	i	22	0.682	0.101	0.201
home	drugs	n	110	0.664	0.028	0.081
sport	computer	i	89	0.607	0.031	0.153
sport	computer	n	445	0.600	0.027	0.105
home	culture	i	105	0.686	0.042	0.090
home	culture	n	525	0.608	0.020	0.078
sport	culture	i	332	0.506	-0.014	0.119
sport	culture	n	1660	0.619	0.009	0.101
drugs	culture	i	25	0.680	0.077	0.190
drugs	culture	n	125	0.584	0.023	0.115
computer	culture	i	144	0.694	0.058	0.114
computer	culture	n	720	0.640	0.032	0.107

Table 4: Results across the subforum duplets. Listed: whether the pair of users actively **i**nteracts or **n**ot (**type**); total number of **pairs** in the sample; proportion of pairs for which $\Delta_{before} - \Delta_{after}$ is **positive**; average **change** $\Delta_{before} - (\Delta_{after})$ and the corresponding **IQR**. Shaded are rows where sample size is smaller than 20 pairs (considered unreliable)

Subforum1	Subforum2	Δ_{pos}	Δ_{change}	Outcome	Comment	$p(\Delta_{pos})$	$p(\Delta_{change})$
home	sport	0.304	0.045	3	div. for non-int.?	**0.002**	**0.013**
computer	drugs	-	-	-	sample too small	-	-
sport	drugs	0.066	0.013	1 or 2		0.180	0.177
home	computer	0.013	0.031	1 or 2		0.485	0.134
home	drugs	0.018	0.073	1		0.519	**0.001**
sport	computer	0.007	0.004	2	small diff.	0.491	0.409
home	culture	0.078	0.022	1		0.076	**0.013**
sport	culture	-0.113	-0.023	?	div. for int.?	1.000	0.994
drugs	culture	0.096	0.054	1		0.242	**0.018**
computer	culture	0.054	0.026	1		0.121	**0.037**

Table 5: Classification of outcomes (see Table 1) per duplet (see Table 4). Δ_{pos} = difference between the proportions of presumably accommodating pairs for interacting and non-interacting users (column **positive** in Table 4). Δ_{change} = difference between the average distance changes for interacting and non-interacting users (column **change** in Table 4). p-values are significance values obtained by bootstrapping (those below 0.05 are boldfaced). Positive values of Δs and small values of ps indicate Outcome 1.

proportion of pairs which (presumably) accommodate is larger than 0.5 in 19 cases out of 19 (though only marginally so for Sport-Culture-i). The average change is positive in 17 cases out of 19 (but note that IQR is very large in most cases, which means considerable variation across pairs).

We compare the observed results with the possible outcomes in Table 5. We concentrate on the effect size and the robustness of effect (how often the same pattern can be observed across duplets and thresholds) rather than statistical significance testing (see Wasserstein et al. (2019) about the limitations and pitfalls of this approach in general and Koplenig (2019) in corpus linguistics in particular). Nonetheless, we also calculate p-values to estimate how likely it is that the observed (or larger) differences between interacting and non-interacting pairs could have arisen by chance. We use a bootstrapping method: we randomly divide all pairs into two samples of the same sizes as the samples of interacting and non-interacting pairs 10,000 times and calculate the proportions of cases when Δ_{pos} and Δ_{change} are larger than or equal to actual values.

Out of nine duplets with sufficient sample size, seven demonstrate the effects which are compatible with either Outcome 1 (overall convergence to a community norm and pairwise accommodation on top of that) or Outcome 2 (just overall convergence) in Table 1. If we use the conventional 0.05 threshold for the p-values, then for four duplets (Home-Drugs, Home-Culture, Drugs-Culture, Computer-Culture) at least one of the two p-values is significant. We judge these four duplets to be most compatible with Outcome 1. In the Sport-Computer duplet, the differences are small, while p-values are large, which indicates no difference between interacting and non-interacting pairs, i.e. Outcome 2. For Sport-Drugs and Home-Computer, the differences are rather large, but the p-values are above the threshold, which makes it difficult to choose between Outcome 1 and Outcome 2. In the Home-Sport duplet, there is a clear difference, but the average distance change for non-interacting users is negative, suggesting divergence. The proportion of converging pairs is, however, marginally larger than 0.5. We label this case as Outcome 3: no clear effect for non-interacting users, thus no evidence for convergence to a community norm. Finally, the Sport-Culture duplet exhibits an unexpected effect: the non-interacting users seem to accommodate, while the interacting users do not (according to the proportion measures) or even diverge (according to the average change).

4 Discussion

From Section 3 it is clear that not all the results unambiguously point in the same direction. It is, however, obvious, that in most cases distance does become shorter, that is, users do converge. Negative results (distance becomes longer) are not only less frequent, but also weaker than most of the positive ones.

By comparing distance changes with the average distance between two different users we show that the effect sizes can be viewed as considerable.

The shortening trend tends to be stronger and more robust for actively interacting pairs, but in some cases there is not enough evidence to prefer Outcome 1 over Outcome 2.

More direct insight into the process of convergence would of course be desirable before it can be stated with certainty that it is *caused* by interactions. Nonetheless, our results provide evidence that it actually *can be so*. In other words, we show that convergence can exist (a necessary condition is meant: distance changes are observed), but not that it definitely exists.

Note that while a reversed causal link can be suggested: users who have similar writing styles will interact more often, or "birds of a feather flock together" (McPherson et al., 2001), it can hardly explain our results on its own: why would users who write on the same subforum and especially those who interact become linguistically closer over time?

There are several reasons to why our results are not as clean as one might want them to be (apart from the obvious "random noise"). First, users in the pairs that we label as "non-interacting" can still interact in other Flashback subforums. Second, while we showed that Cosine Delta is a very good measure for linguistic distance, the definition of an interaction is more arbitrary. There is already a tradition of using the "post-nearby-in-the-same-thread" measure (Hamilton et al., 2017; Del Tredici and Fernández, 2018), but it has not really been evaluated. Overall, further exploration of the same (or similar) data is of course desirable. Different experimental designs, different thresholds, different measures would show how robust the observed effects are.

We find the following questions particularly appealing for future studies.

- If we compare accommodation across interacting pairs, will it be correlated with the number/intensity of interactions?
- What happens if we consider not only direct connections between users, but also indirect ones? If A interacts with B, B interacts with C, but A does not directly interact with C: do A and C become closer?
- What happens if A and C from the previous example are pulling the style of B into different directions?
- Why do we sometimes observe negative values that suggest divergence (the distance increases)? Danescu-Niculescu-Mizil et al. (2013) observe an increasing divergence between the community norm and the production of a user who is become less active in the community (and will eventually leave), but it is unclear whether this can explain our results.
- Is it possible to explain convergence and divergence better if we take into account the content of the users' posts and the relationship between users?

5 Conclusions

We show that writing styles of users who participate in the same subforums do become more similar over time and that this increase in similarity tends to be stronger for pairs of users who actively interact (compared to those who do not interact), though this is not an exceptionless trend. These results support the accommodation hypothesis (let us repeat Labov's wording: "the more often people talk to each other, the more similar their speech will be").

It is desirable to see if the observed effects can be replicated in similar studies with different experimental settings.

All data and scripts necessary to reproduce the study are openly available.[4]

[4] https://github.com/AleksandrsBerdicevskis/LinguisticConvergence

Acknowledgements

This work has been supported by the Cassandra project (funded by Marcus and Amalia Wallenberg Foundation, donation letter 2020.0060) and supported by the Swedish national research infrastructure Nationella språkbanken, funded jointly by the Swedish Research Council (2018–2024, contract 2017-00626) and the 10 participating partner institutions.

References

Allan Bell. 1984. Language style as audience design. *Language in Society*, 13(2):145–204.

Johanna Björklund and Niklas Zechner. 2017. Syntactic methods for topic-independent authorship attribution. *Cambridge University Press*.

Lars Borin, Markus Forsberg, and Johan Roxendal. 2012. Korp — the corpus infrastructure of språkbanken. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 474–478, Istanbul, Turkey. European Language Resources Association (ELRA).

John Burrows. 2002. 'Delta': a Measure of Stylistic Difference and a Guide to Likely Authorship. *Literary and Linguistic Computing*, 17(3):267–287.

Patricia Cukor-Avila and Guy Bailey. 2001. The effects of the race of the interviewer on sociolinguistic fieldwork. *Journal of Sociolinguistics*, 5(2):252–270.

Cristian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. No country for old members: User lifecycle and linguistic change in online communities. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 307–318, New York, NY, USA. Association for Computing Machinery.

Marco Del Tredici and Raquel Fernández. 2018. The road to success: Assessing the fate of linguistic innovations in online communities. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1591–1603, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Penelope Eckert. 2012. Three waves of variation study: The emergence of meaning in the study of sociolinguistic variation. *Annual Review of Anthropology*, 41(1):87–100.

Stefan Evert, Thomas Proisl, Thorsten Vitt, Christof Schöch, Fotis Jannidis, and Steffen Pielström. 2015. Towards a better understanding of Burrows's delta in literary authorship attribution. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 79–88, Denver, Colorado, USA. Association for Computational Linguistics.

Cynthia Gallois, Howard Giles, Elizabeth Jones, Aaron C Cargile, and Hiroshi Ota. 1995. Accommodating intercultural encounters: Elaborations and extensions. In Richard L. Wiseman, editor, *Intercultural Communication Theory*, pages 115–147. Sage Publications.

Simon Garrod, Alessia Tosi, and Martin J Pickering. 2018. Alignment during interaction. In Shirley-Ann Rueschemeyer and M. Gareth Gaskell, editors, *The Oxford Handbook of Psycholinguistics, 2 ed.* Oxford University Press.

Howard Giles. 1973. Accent mobility: A model and some data. *Anthropological Linguistics*, 15(2):87–105.

William Hamilton, Justine Zhang, Cristian Danescu-Niculescu-Mizil, Dan Jurafsky, and Jure Leskovec. 2017. Loyalty in online communities. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):540–543.

Internetstiftelsen. 2021. Svenskarna och internet 2021. https://svenskarnaochinternet.se/rapporter/svenskarna-och-internet-2021/.

Fotis Jannidis, Steffen Pielström, Christof Schöch, and Thorsten Vitt. 2015. Improving Burrows' Delta. an empirical evaluation of text distance measures. In *Digital Humanities Conference*, volume 11, Sydney.

Alexander Koplenig. 2019. Against statistical significance testing in corpus linguistics. *Corpus Linguistics and Linguistic Theory*, 15(2):321–346.

William Labov. 2001. *Principles of linguistic change. Volume 2: Social factors*. Blackwell, Oxford.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444.

George K. Mikros and Eleni K. Argir. 2007. Investigating topic influence in authorship attribution. In *SIGIR 2007 Workshop: Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection*, pages 29–36.

Aurélie Nardy, Jean-Pierre Chevrot, and Stéphanie Barbu. 2014. Sociolinguistic convergence and social interactions within a group of preschoolers: A longitudinal study. *Language Variation and Change*, 26(3):273–301.

Dong Nguyen and Carolyn P. Rosé. 2011. Language use as a reflection of socialization in online communities. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pages 76–85, Portland, Oregon. Association for Computational Linguistics.

Scott A. Reid and Howard Giles. 2008. Social identity theory. In *The International Encyclopedia of Communication*. John Wiley & Sons, Ltd.

John R Rickford, Faye McNair-Knox, et al. 1994. Addressee-and topic-influenced style shift: A quantitative sociolinguistic study. In *Sociolinguistic perspectives on register*, pages 235–276. Oxford University Press.

Peter W. H. Smith and W. Aldridge. 2011. Improving authorship attribution: Optimizing Burrows' Delta method*. *Journal of Quantitative Linguistics*, 18(1):63–88.

Ronald L. Wasserstein, Allen L. Schirm, and Nicole A. Lazar. 2019. Moving to a world beyond "$p < 0.05$". *The American Statistician*, 73(sup1):1–19.

Rules and neural nets for morphological tagging of Norwegian – Results and challenges

Dag Trygve Truslew Haug[1,2], **Ahmet Yıldırım**[1], **Kristin Hagen**[2], **Anders Nøklestad**[2]
[1]Department of Linguistics and Scandinavian Studies, University of Oslo
[2]Humit, University of Oslo
{daghaug,ahmetyi,kristiha,noklesta}@uio.no

Abstract

This paper reports on efforts to improve the Oslo-Bergen Tagger for Norwegian morphological tagging. We train two deep neural network-based taggers using the recently introduced Norwegian pretrained encoder (a BERT model for Norwegian). The first network is a sequence-to-sequence encoder-decoder and the second is a sequence classifier. We test both these configurations in a hybrid system where they combine with the existing rule-based system, and on their own. The sequence-to-sequence system performs better in the hybrid configuration, but the classifier system performs so well that combining it with the rules is actually slightly detrimental to performance.

1 Introduction

The Oslo-Bergen Tagger (OBT, Hagen and Johannessen 2003; Johannessen et al. 2012) is a widely used tool for morphological tagging of Norwegian text. It has existed in various incarnations for around 25 years, first as a purely rule-based system and later coupled with a statistical module for disambiguation. In this paper, we report on our recent efforts to bring the system into the age of neural networks. The question that arises is whether combining the neural system with the existing rules will boost accuracy over a purely neural system. We build two kinds of neural net configurations, one encoder-decoder transformer framework (sequence-to-sequence, seq2seq) and one sequence classification (seqClass) approach. We show that there are challenges in combining rules and neural nets due to divergent tokenisations when the seq2seq approach is employed. In the seqClass approach, on the other hand, the neural net performs so well that combining it with the rule-based approach is actually detrimental to performance, showing that rule-based methods are not required in the morphological tagging of a language like Norwegian, where a large language model is available (NbAiLab, 2021) and there is sufficient labeled data for fine-tuning. However, we still believe that the rule-based system can be useful for lemmatisation and compound analysis, which we do not consider here.

The structure of the paper is as follows: In section 2 we give some historical background on OBT and in section 3 we describe the current status of its rule-based component. Section 4 describes the training and evaluation data that we have used in developing the new systems. Section 5 then provides the details of how our neural systems were trained. Section 6 describes how they were combined with the rule system. Section 7 evaluates the performance of each of the neural systems on their own as well as in combination with the rules. Section 8 concludes.

2 History of the Oslo-Bergen Tagger

The Oslo-Bergen Tagger was originally developed between 1996 and 1998 by the Tagger Project at the University of Oslo. Rules for morphological and syntactic disambiguation were written in the first version of the Constraint Grammar framework (Karlsson et al., 1995), retrospectively called CG1. The input to CG disambiguation rules is multitagged text, i.e., text where each token has been annotated with all possible lexical analyses. Hence, the project also developed a lexicon with lemmas and inflected forms (later known as Norsk ordbank[1]) and a combined tokeniser/multitagger. The tagger was developed for both Bokmål and Nynorsk, the two written varieties of Norwegian. In this article, we will only focus on the Bokmål

[1]https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-5/

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 425–435

version of the tagger, and only on the tokeniser and the morphological disambiguation.

On an unseen evaluation corpus with a wide variety of text genres, OBT achieved an F1-score of 97.2 (Hagen and Johannessen, 2003, 90), with a precision of 95.4 and a recall of 99.0. At the time, this was considered acceptable as the tagger was mostly used to annotate written corpora for linguistic research, where a high recall was considered more important than a high precision.

Since 1998, both the tokeniser and the CG rule interpreter have been changed or modernised several times by different projects (Johannessen et al., 2012). The latest version has an updated lexicon and a tokeniser written in Python which in most cases mirrors the original tokeniser, with the major exception that multiword expressions like *blant annet* ('among other things' - adverb) have been removed from the lexicon and are now dealt with in the CG module. The CG1 rules have been translated to the more efficient and expressive CG3 format and are used with a rule interpreter made by the VISL project at the University of Southern Denmark. Remaining morphological ambiguities and lemma disambiguation are dealt with by a statistical module, implemented as a Hidden Markov Model. This OBT+Stat system achieved an accuracy of around 96% (Johannessen et al., 2012).

3 The rule-based tokeniser and tagger

In this section, we first present some of the main tasks for the tokeniser and multitagger before we give a short description of the constraint grammar module. The multitagger uses a lexicon with all possible lexical readings, where a reading is a combination of a lemma and a morphosyntactic tag chosen from a set of 149 possible analyses.[2] The main principle for tokenisation is to split tokens on blank space or a sentence delimiter like a full stop or a question mark. For each token identified, the original word form is looked up in the lexicon. Non-sentence initial capitalised words are identified as proper nouns, while other words that exist in the lexicon are assigned all readings found there. If the word is not found in the lexicon and is not identified as a proper noun, the word is sent to a compound analyser. Most unknown words will get an analysis here, as many of them are productively created compounds. Some words will still get the tag *ukjent* ('unknown') from the tokeniser. These words are often dialect words not standardised in the lexicon or foreign words. Figure A in the Appendix shows how the tokeniser and multitagger deals with the sentence *TV-programmet "Ut i naturen" begynner kl. 21.15.* ('The TV program "Ut i naturen" starts at 21.15.'), which has quotation marks, abbreviations, and a time expression.

The tokeniser also identifies sentence boundaries using sentence delimiters, a list of known abbreviations and linguistic rules. Headlines are identified by rules as well and get their own tag.

The constraint grammar module takes tokenised and multitagged text as input and its main task is to reduce the number of readings to ideally one per word. The number of readings left by the multitagger varies a lot. In the test corpus used in this article (which will be further described in Section 4) there are on average 2,04 readings per word. After the CG rules are applied, there are on average 1,09 readings left per word.

Figure B in the Appendix shows the output from the CG module in debug mode for the sentence *Rosa cupcakes hører kanskje med når man skal ha bloggtreff?* ('Pink cupcakes might be part of a blog meeting?'). Readings that have been removed start with ";" and the ID numbers of the rules applied are appended to each reading. Note that the English loan word *cupcakes* is not identified in the lexicon or in the compound analyser and has got the tag *ukjent* 'unknown'. The compound *bloggtreff* 'blog meeting' was not in the lexicon but has got two readings from the compound analyser. As the examples show, there are both REMOVE rules (remove a reading) and SELECT rules (select a reading). A rule can be very simple, like rule 2430 in Figure 1 that says "select the verb infinitive reading if the verb to the left is a modal auxiliary and not in the set of dangerous infinitives (= not likely infinitives)".

```
#:2430
SELECT:2430 (verb inf) IF
(NOT 0 farlige-inf)
(-1 m-hj-verb)
;
```

Figure 1: Simple SELECT rule

Figure 2 shows an example of a more complex rule with linked context conditions somewhere to the right in the sentence. The rule says: "choose

[2]The complete list is available at `http://tekstlab.uio.no/obt-ny/morfosyn.html`.

the subjunction reading – if somewhere to the right there is a safe noun or pronoun (stop looking if a word on the way has a reading that is not an adverb, adjective or determinative) – and – if there is a word in the present or past tense after the noun/pronoun (adverbs between are fine)."

```
#:2579
SELECT:2579 (sbu) IF
(...)
(**1C subst/pron BARRIER
   ikke-adv-adj-det)
(**1C subst/pron LINK *1
   ikke-adv LINK 0 pres/pret)
;
```

Figure 2: More complex SELECT rule

The CG grammar for Bokmål has more than 2300 rules. 1995 of them are SELECT rules. Some rules apply to all possible words, while some are rules for specific word forms. REMOVE rules look the same as SELECT rules but remove a reading instead of selecting it. During development, we checked the impact of each rule on the recall and precision on a training corpus of 100 000 words from novels, newspapers and magazines before it was added to the rule set.

4 Training and evaluation data

The training and evaluation corpora that were used in earlier stages of development of the OBT system are no longer suitable because the tagset and the tokenisation principles have evolved. Instead of bringing this corpus up to date, we chose to use the Norwegian Dependency Treebank (NDT, Solberg et al. 2014) in the development of the latest version of OBT. The Bokmål part of NDT is around 300 000 tokens and consists of blog text, news text, parliament proceedings and government white papers. A problem that we only later became aware of is that most of the raw text contained in the NDT probably went into the Norwegian BERT encoder that we use in our machine learning experiment, which may have caused some data leakage, even if the model did not see the tagged text.

While the principles for annotation in NDT and OBT are close, there are still differences in detail. To ensure that the annotations were as aligned as possible, we ran OBT without statistical disambiguation on the pure text of the NDT and compared the output to the NDT annotations. If the NDT analysis was not among the analyses produced by OBT, we either corrected the NDT annotation if that was the source of the error, or changed the rules of the OBT system if that could easily be done. This process was iterated a few times. The goal was to improve the quality of the training data for the neural component and to align the output of the OBT with the NDT as the annotation guidelines were slightly different. Also, since the plan was to combine OBT with a neural system, ambiguity reduction by OBT was not a goal in itself if we thought the ambiguity could be resolved by the neural component. Notice that during this period, the whole data set was used for development, which inflates the performance of the rules (and hence the hybrid system we discuss later on) somewhat. But in practice, relatively few changes were made and we did not achieve a full alignment of the annotation guidelines.

The performance of the rule-based system by the end of this phase is shown in Table 1. We see that OBT produces a correct and unambiguous analysis for 90.7% of the tokens and only (one or more) incorrect analyses for 1.8% of the tokens. For 7.5% of the tokens, OBT produces an ambiguous analysis containing the correct tag as one possibility, and the role of the statistical system in a hybrid setup is to pick the correct analysis in these cases. The results are noticeably different from those reported during testing in the nineties (see Section 2), probably because we were not able to fully align the annotation principles of OBT and NDT, and because the precision was calculated differently back then (for example, both the masculine and the feminine reading were regarded as one correct reading for ambiguous feminine and masculine nouns in unmarked contexts).

result	freq	
unambiguous correct	280650	(90.7%)
ambiguous incl. correct	23219	(7.5%)
wrong	5413	(1.8%)

Table 1: Performance of the rule-based system

Finally, for the neural systems, we split the corpus into train-dev-test sets. While doing this, we applied a simple process for making sure the output tags in the training set covered all output tags in the dev and test sets. The aim is to ensure that the model was trained with samples from all tags. We do this by, first, initializing the Python random seed as 0, then, splitting the data and checking if

the training set covers all tags. If it does not, we increase the random seed by one and do the same until we find a training set that covers all the tags in the other sets. This way, we randomly split the dataset into 80-10-10 percent partitions to obtain train-dev-test datasets respectively.

5 The neural systems

Recently, a BERT (Devlin et al., 2018) pre-trained encoder (nb-bert-base) was published by the Norwegian National Digital Library (Kummervold et al., 2021). This pre-trained encoder for Norwegian provides a rich feature set that was previously lacking for the language. Furthermore, since the tagged corpus is very small in comparison to the corpus the pre-trained model was trained on, it is important to use the pre-trained model to be able to generalise to unseen data. We use two different neural system configurations that incorporate this encoder.

5.1 The seq2seq configuration

With this configuration, we follow an approach similar to that of Omelianchuk et al. (2020) to tag the sentences using the pre-trained model. Seq2seq models have two main components: an encoder and a decoder. The encoder side is set as the encoder nb-bert-base (NbAiLab, 2021). For the decoder, we randomly initialise 6 layers of size 768 with 12 attention heads. The decoder also has cross-attention layers as it was shown to be effective in seq2seq training (Gheini et al., 2021). We freeze the encoder weights throughout the training since using the encoder as a feature extraction mechanism in this way was shown to be beneficial (Zoph et al., 2016) and is a common practice (Gheini et al., 2021). We use the EncoderDecoderModel provided by the HuggingFace transformers library (Wolf et al., 2020) to configure and train a model.

The encoder-decoder model gets as its input the identifiers of the tokens (token numbers) in the input vocabulary and outputs the token numbers in the output vocabulary. Thus, the input and output are tokenised using these vocabularies. Since the encoder model had already been trained (nb-bert-base) using the widely-utilised sub-word tokeniser Wordpiece (Wu et al., 2016), we use that tokeniser as provided by the Huggingface Tokenizers library. For the decoder side, since our vocabulary size is very small and obvious (82 tags and 5 extra special tokens such as [CLS] and [SEP]), we do not need to train a special tokeniser. We define the vocabulary manually with these output tokens for use by the Wordpiece tokeniser.

The data were formatted to train the seq2seq network. Figure 3 shows an example of input and output for a sentence. The input is the tokenised form of the sentence. The output is the sequence of serialised tags for each token in the input. The token <next_token> is an indicator that all tags of the corresponding input token have finished and tags of the next input token start afterward.

```
INPUT: Men det er bare noe jeg tror .
OUTPUT:
:konj: clb <next_token>
:pron: 3 ent nøyt pers <next_token>
:verb: pres <next_token>
:adv: <next_token>
:pron: 3 ent nøyt pers <next_token>
:pron: 1 ent hum nom pers <next_token>
:verb: pres <next_token>
$punc$ :clb: <punkt>
```

Figure 3: A sample of sentence input and output for seq2seq training.

The training configuration is as follows: We use the Adam optimiser (Kingma and Ba, 2015) with a learning rate of 0.0001. We set the batch size to 16 sentences as this is the amount the graphic cards could handle. We use the negative log-likelihood loss (Yao et al., 2020) to compute the loss in each batch between the model output and the expected output. For any other parameter not mentioned in this section, we use the default value defined by version 4.17.0 of the Transformers library in the objects of the following types: BertConfig, EncoderDecoderModel, EncoderDecoderConfig, and BertModel.

We evaluate the model using the dev set during the training. We do this by using the BLEU score (Papineni et al., 2002) that is widely utilized to evaluate seq2seq models. We compute the BLEU score between the expected output and the model output for each sentence. We get the average of these scores for the whole dev set. We run the training for 300 epochs and keep the model that results in the maximum average BLEU score for the dev set. To combine the model output with the rules, we use the model's .generate() function (HuggingFace, 2023a) implemented by the library. We set the early_stopping, return_dict_in_generate, and output_scores param-

eters to True. We set num_return_sequences and num_beams to 10 to get the 10 most probable readings given a sentence.

5.2 The seqClass configuration

Sequence classification – also referred to as token classification – is a method used to classify a sequence (one or more tokens) into one or more classes such as the type (person, organisation) or sentiment (positive, negative). BERT models have multiple layers that are pre-trained on the language. The Norwegian pre-trained BERT outputs 12 layers (also called hidden states) where each is a 768-dimension vector for each token. Thus, this output is input to a classifier to classify each token. The HuggingFace transformers library (HuggingFace, 2023b) provides a token classification framework that can be used for this purpose. It adds a linear layer on top of the hidden states to make sequence classification possible using the pre-trained encoder.

The dataset has 82 different tags which are used together in different combinations. We observe that the training set has 327 different uses of these combinations.[3] Thus, we treat each combination as a class for this computation. We classify each token into one of these classes which indicates a tagset for that token. We do this by labeling each class as a sequence of zeros and ones where each digit corresponds to one tag. Figure 4 shows an example of tokens and classes of those tokens, where the length of the class names (0's and 1's), which is really 82, is shortened to fit into the figure. The position of 1's in the string indicates the tag assigned to the token. For example, for the first token "Men" the first two columns are assigned which indicate the ":konj:" and "clb" tags (see also Figure 3 for tags of this sentence).

```
Men    11000000000000000000000000000000000
det    00111110000000000000000000000000000
er     00000001100000000000000000000000000
bare   00000000010000000000000000000000000
noe    00111110000000000000000000000000000
jeg    00001000001110000000000000000000000
tror   00000001100000000000000000000000000
.      00000000000001110000000000000000000
```

Figure 4: A sample of a sentence's tokens and their classes used in sequence classification.

[3]In addition to the 149 morphosyntactic analyses (see footnote 2), this includes combinations with various tags that do not convey morphosyntactic information and are ignored during evaluation.

Throughout the training, we use the default parameters defined in the library. We use the Adam optimiser (Kingma and Ba, 2015) with an adaptive learning rate starting from 0.00005. The library uses Cross Entropy Loss (Zhang and Sabuncu, 2018) and picks the model that performs best on the dev set by computing an F1 score. We check the dev set performance for each epoch and run the training for 30 epochs. When combining the model output with the rules, we use the unnormalized final scores of the model (logits) and use torch.topk() to get the topmost 10 probable readings given a sentence.

6 Combining neural nets and rules

To combine the output of a neural tagger with the CG tagger, we need to find the intersection of the tag assignments produced by the two taggers. Ideally, we would be able to find such intersections for each individual token separately. However, since the probability of a reading for a particular token depends on the selected readings for all other tokens in the sentence, the only viable option is to consider readings for entire sentences. Thus, for each input sentence, we extract the ten most probable tag assignments produced by the network. Then, for each reading in this list, ordered by decreasing probability, we go through each token and check whether the tag assigned by the network is also found among those left by the CG disambiguation rules. If it is not found, we skip to the next reading in the list. If it is found, we go on to check the next token, and so on until we reach the end of the sentence, at which point the reading is picked as the selected one for the sentence. When the tokenisations are different, it is not clear what to do. But if the tokens are the same, but the intersection of the sets of possible tags left by the CG system and the neural net is empty, we can default to the most probable reading in the neural net output.

Figure 5 shows a case where the tokenisation of the seq2seq neural system does not match with that of OBT. The neural system has split the initial, unknown proper name at a hyphen, whereas the CG tagger keeps it as one token. Since tokenisation is part of a preprocessing step and misalignments in tokenisation is a problem to be solved separately from tag assignment, we simply disregard such sentences in the evaluation. However, it should be noted that this problem is only acute for

the seq2seq system, which produced mismatching tokenisation in 205 out of 2003 sentences (10.2%). For the seqClass system, the problem is smaller: 57 out of 2003 sentences (2.8%).

```
Neural net: Garosu - gil , som betyr [...]
CG: Garosu-gil , som betyr [...]
```

Figure 5: Mismatching tokenisation

Figure 6 shows the problem of mismatching tags. For the first word, the CG tagger has left five possible analyses, and the neural net has correctly disambiguated to the plural adjective reading. However, OBT did not recognise the second word, *cupcakes*, and has therefore left an *ukjent* ('unknown') tag while the neural system has no analysis with that tag. Notice that the figure only shows the neural system's most probable assignment of tags to the whole sentence. The actual output is a probability distribution over tag assignments, but in this case, no probability was assigned to any tag assignment containing the *ukjent* tag for *cupcakes*, which is the only analysis produced by the rule system.

```
Neural net:
Rosa         adj fl pos
cupcakes     subst appell fl mask ub  <---
hører        verb pres
kanskje      adv
med          prep
når          sbu
man          pron ent hum nom pers
skal         verb pres
ha           verb inf
bloggtreff   subst appell ent nøyt ub
?            clb <spm>
CG:
Rosa         adj fl pos
             adj nøyt ub ent pos
             adj ub m/f ent pos
             subst appell ubøy
             subst appell fem be ent
cupcakes     ukjent                   <---
hører        verb pres
kanskje      adv
med          prep
når          sbu
man          pron ent hum nom pers
skal         verb pres
ha           verb inf
bloggtreff   subst appell ent nøyt ub
             subst appell fl nøyt ub
?            clb <spm>
```

Figure 6: Non-intersecting tags

For such cases, we default to the most probable analysis generated by the neural net. This is not necessarily the best option: as we will see in Section 7, the seq2seq system is often incorrect in cases where the tag assignments do not intersect. Moreover, the problem of mismatching tag assignments is quite common, happening in 386 out of the 2003 sentences (19.3%).

In the seqClass system, non-intersecting tag assignments are even more frequent, at 466 sentences (23.3%). However, as we will see in the next section, the neural net in this configuration is more precise than the rules, so that defaulting to its output yields the correct reading.

system	accuracy
pure seq2seq	92.71
seq2seq + OBT	94.15
pure seqClass	100.0
seqClass + OBT	99.99

Table 2: Accuracy of different systems

7 Evaluation and error analysis

We evaluate both the seq2seq system and the seqClass system on their own and as combined with the rule-based system in the way described in Section 6. This yields four different systems. The performance of the four systems is shown in Table 2.

These numbers are only computed over sentences where the tokenisation matches. This means that the seq2seq system, in both its pure and hybrid form, is tested only on sentences where the seq2seq system, the OBT tagger and the gold agree on the tokenisation. As we saw in section 6, this means that 10.2% of the test data are left out. It would have been possible to test the pure seq2seq system on the sentences where its tokenisation agrees with the gold, without considering what OBT does, but since we want to compare the pure neural system to the hybrid system, we held the evaluation set constant between these two setups. Similarly, for the seqClass system, we left out the 2.8% of sentences where either OBT or the neural system had tokenisation that does not match the gold for both the pure and the hybrid system. Notice that this means the seqClass system is tested on a larger set of sentences than the seq2seq system.

Overall, we see that the seqClass system performs best and in fact achieves a perfect score. This is of course a rather debatable result, which we will look into in section 7.2. But notice that the 2.8% of sentences with diverging tokenisations

are incomparable and therefore not evaluated here, though they could obviously be considered errors of the system.

7.1 The seq2seq system

The seq2seq system performs reasonably well on its own, but clearly benefits from being intersected with the rules, yielding a 1.3% accuracy boost to 94.1%. By contrast, the widely used Spacy tagger reports an accuracy of 95.0% for morphological tagging of Norwegian UD.[4]

Most of the errors in this setup comes from the fact that we default to the best neural analysis when there is no intersection. As it turns out, the neural system is wrong in most of these cases. If we restrict attention to only sentences where the tags intersect (70.5% of the total), accuracy is at 99.0%. Put another way, when we reduce the test set in this way, its size decreases by 8036 tokens from 26648 to 18612, but the number of errors decreases from 1940 to 565. This indicates an error rate of 17.1% on the tokens in sentences where the intersection of the tag assignments from the neural system and the CG tagger is empty.

Turning now to the kind of errors the seq2seq system makes, we show the twelve most common error types of the pure and the hybrid system in Table 3 and Table 4 respectively. We see that the most common error is mixing up the distinction between neuter and common gender adjectives, which in many cases is not expressed morphologically. Other than that, most errors involve either over- or underpredicting the tag :prep: (preposition). This error source is somewhat reined when the system is interfaced with the rules. But many errors of this kind remain, either because this analysis is also suggested by the rules and so picked as the most probable tag, or more likely because there is no intersection between the tag assigments, i.e. neither :prep: nor any other tag suggested by the neural system is among the tags left by the rule-based system.

Overall, this confusion around the :prep: tag seems a distinct deficit of the seq2seq model. Other errors, such as those involving gender, or the number of indefinite neuter nouns (which make no morphological singular/plural distinction), or the identification of perfect participles which often co-exist with homonymous adjectives in Norwegian (as in other Germanic languages, cf. English ‘bored’) are more as one would expect from any system because there might not be enough signal in the training data to pick up the distinctions, which often depend on subtle properties of the context. However, what we observe here is that intersecting with the rules actually worsens the accuracy. The hybrid system overapplies the adjective analysis in two different varieties for a total of 17+13 errors. By contrast, in the pure seq2seq setup, this error is not frequent enough to figure in the table. It does occur in 21 cases, but that is still notably less than in the hybrid system. This shows that the CG rules wrongly disambiguate these cases, which is not surprising since the distinction as made according to the NDT guidelines relies on semantic distinctions. It would be hard to tune the CG rules to those distinctions, and we did not make any attempt at that, but there seems to be enough signal in the data for the seq2seq system to pick it up to some extent.

7.2 The seqClass system

The seqClass system achieves a suspicious, perfect score on our test set when used alone, and makes one error when combined with the rules. This error is instructive in itself: it concerns a single-word "sentence", namely the heading "Justisdepartementet" (‘The Department of Justice’). The CG tagger considers this a common noun. This is only the fifth most probable tag according to the neural net, but it is among the possibilities and so it is chosen by the hybrid system, although the gold considers it a proper noun.

This is the only instance of such an error. In other words, the seqClass system not only assigns the highest probability to the correct tag in every case, but also performs well enough as to not rank any incorrect suggestions by the CG system among the top 10 readings that we consider for intersection, except in this one case.

The looming question is of course how the system manages to perform so well. Some degree of overfitting must have taken place, but can hardly explain everything. Moreover, as we noted in Section 4, it is likely that all or at least most of the raw text of NDT went into the Norwegian BERT model, which may have caused some data leakage. More worryingly, we cannot completely exclude the possibility that the language model has been

[4] See `https://spacy.io/models/nb`. As the Norwegian UD corpus (Øvrelid and Hohle, 2016) is an automatic conversion of the NDT corpus, the complexity of the tasks should be comparable, although the test split is not identical.

Gold tag	Predicted tag	Freq
[':adj:', 'ent', 'nøyt', 'pos', 'ub']	[':adj:', 'ent', 'm/f', 'pos', 'ub']	24
[':subst:', 'appell', 'ent', 'mask', 'ub']	[':prep:']	24
[':prep:']	[':subst:', 'appell', 'ent', 'mask', 'ub']	19
[':verb:', 'pres']	[':prep:']	18
[':subst:', 'appell', 'ent', 'mask', 'ub']	[':subst:', 'appell', 'ent', 'fem', 'ub']	18
[':prep:']	[':subst:', 'prop']	17
[':subst:', 'appell', 'ent', 'fem', 'ub']	[':subst:', 'appell', 'ent', 'mask', 'ub']	16
[':prep:']	['$punc$', ':<komma >:']	15
[':prep:']	[':verb:', 'pres']	14
[':subst:', 'appell', 'fl', 'mask', 'ub']	[':subst:', 'appell', 'fem', 'fl', 'ub']	14
[':subst:', 'mask', 'prop']	[':subst:', 'prop']	14
[':subst:', 'appell', 'ent', 'mask', 'ub']	[':subst:', 'appell', 'ent', 'nøyt', 'ub']	14

Table 3: Most frequent errors, pure seq2seq system

Gold tag	Predicted tag	Freq
[':adj:', 'ent', 'nøyt', 'pos', 'ub']	[':adj:', 'ent', 'm/f', 'pos', 'ub']	22
[':subst:', 'appell', 'ent', 'mask', 'ub']	[':subst:', 'appell', 'ent', 'fem', 'ub']	18
[':subst:', 'appell', 'ent', 'mask', 'ub']	[':prep:']	18
[':prep:']	[':subst:', 'appell', 'ent', 'mask', 'ub']	17
[':verb:', 'pres']	[':prep:']	17
[':verb:', 'perf-part']	[':adj:', '<perf-part>', 'ent', 'm/f', 'ub']	17
[':prep:']	[':subst:', 'prop']	16
[':verb:', 'perf-part']	[':adj:', '<perf-part>', 'ent', 'nøyt', 'ub']	13
[':subst:', 'appell', 'fl', 'mask', 'ub']	[':subst:', 'appell', 'fem', 'fl', 'ub']	13
[':prep:']	[':verb:', 'pres']	12
[':subst:', 'appell', 'ent', 'nøyt', 'ub']	[':subst:', 'appell', 'fl', 'nøyt', 'ub']	12
[':verb:', 'perf-part']	[':prep:']	11

Table 4: Most frequent errors, hybrid seq2seq system

exposed to the CONLL file (and hence the manually corrected tags), although it seems unlikely. In any case, we would have expected some errors in the tokens where we changed the analysis. Moreover, none of these factors would explain why the model is also able to assign a very low probability to the incorrect suggestions from the CG.

We plan to conduct a more thorough test of the system on recent text which the BERT model cannot have been exposed to. So far we have only been able to conduct a very preliminary test. We downloaded web text from nrk.no (the Norwegian national broadcaster) from 2023, i.e. after the Norwegian BERT was published. This text was tagged both with the original system of OBT + HMM-based disambiguation, and with the new seqClass system. For the first 2000 tokens, we inspected all mismatches between the two systems, on the (questionable) assumption that whenever the two systems agree, the tag is likely correct. We found 144 discrepancies, and by manual judgement 137 were considered errors by the old system, and 7 were considered errors by the pure seqClass system. This evaluation method is obviously not perfect, but it does suggest that the pure seqClass system makes very few errors. Further proper evaluation must follow, but the results are clear enough to discourage future work on the rule-based system.[5]

8 Conclusion

We have presented our efforts to improve the Oslo-Bergen tagger for Norwegian morphological tagging. Two neural systems were trained, based on a sequence-to-sequence setup and a sequence classifier setup, both built on top of the Norwegian BERT model of Kummervold et al. (2021). Both were tested on their own and in combination with the rule-based OBT system. The sequence-to-sequence system did not outperform earlier benchmarks on its own, but improved when combined the rules. However, the sequence classification setup was much better and in fact achieved a surprising perfect score on the test set. While we will explore the causes of this, preliminary testing on new data supports the conclusion that the new system makes very few errors, and we will focus on validating this in a more proper evaluation setting.

[5]The seqClass model is available for download at `https://github.com/textlab/norwegian_ml_tagger`.

Acknowledgments

This work was supported by the CLARINO project, funded by RCN FORINFRA grant no. 295700, and the Universal Natural Language Understanding project, funded by RCN IKTPLUSS grant no. 300495.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. http://arxiv.org/abs/1810.04805.

Mozhdeh Gheini, Xiang Ren, and Jonathan May. 2021. Cross-attention is all you need: Adapting pretrained Transformers for machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1765, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kristin Hagen and Janne Bondi Johannessen. 2003. Parsing nordic languages (panola) – norsk versjon. In Henrik Holmboe, editor, *Nordisk Sprogteknologi 2002*, pages 89–96. Museum Tusculanum, Copenhagen.

HuggingFace. 2023a. Generation. https://huggingface.co/docs/transformers/v4.28.0/en/main_classes/text_generation#transformers.GenerationMixin.generate, Accessed: 16.04.2023.

HuggingFace. 2023b. Token classification. https://github.com/huggingface/transformers/tree/main/examples/pytorch/token-classification, Accessed: 05.04.2023.

Janne Bondi Johannessen, Kristin Hagen, André Lynum, and Anders Nøklestad. 2012. Obt+stat. In Gisle Andersen, editor, *Exploring Newspaper Language: Using the web to create and investigate a large corpus of modern Norwegian*, pages 51–66. John Benjamins, Amsterdam.

Fred Karlsson, Atro Voutilainen, Juha Heikkilä, and Arto Anttila, editors. 1995. *Constraint Grammar: A Language-Independent Framework for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Per E Kummervold, Javier De la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. 2021. Operationalizing a national digital library: The case for a Norwegian transformer model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 20–29, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

NbAiLab. 2021. Norwegian Transformer Model. https://github.com/NbAiLab/notram/tree/0c90d6b28008df514c4ac847e4c9d68f4709a181, Accessed: 12.12.2022.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the 15th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163—-170. Association for Computational Linguistics.

Lilja Øvrelid and Petter Hohle. 2016. Universal dependencies for norwegian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1579–1585.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Per Erik Solberg, Arne Skjærholt, Lilja Øvrelid, Kristin Hagen, and Janne Bondi Johannessen. 2014. The Norwegian dependency treebank. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 789–795, Reykjavik, Iceland. European Language Resources Association (ELRA).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Hengshuai Yao, Dong-lai Zhu, Bei Jiang, and Peng Yu. 2020. Negative log likelihood ratio loss for deep neural network classification. In *Proceedings of the Future Technologies Conference (FTC) 2019*, pages 276–282, Cham. Springer International Publishing.

Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

Appendix: sample multitagger and CG output

```
<word>Tv-programmet</word>
"<tv-programmet>"
"tv-program" subst appell nøyt be ent
 samset-leks <*program> <+programmet>
<word>«</word>
"<«>"
"$«" <anf>
<word>Ut</word>
"<ut>"
"ut" prep
"ut" adv
<word>i</word>
"<i>"
"i" prep
"i" subst appell mask ub ent
<word>naturen</word>
"<naturen>"
"natur" subst appell mask be ent
<word>»</word>
"<»>"
"$»" <anf>
<word>begynner</word>
"<begynner>"
"begynne" verb pres
"begynner" subst appell mask ub ent
<word>kl.</word>
"<kl.>"
"kl." subst appell fork
<word>21.15</word>
"<21.15>"
"21.15" subst <klokke>
"21.15" det kvant
<word>.</word>
"<.>"
"$." clb <<< <punkt> <<<
```

Figure A: Tokenised and multitagged sentence

```
<word>Rosa</word>
"<rosa>"
"rosa" adj fl pos
"rosa" adj nøyt ub ent pos
"rosa" adj ub m/f ent pos
"rosa" subst appell ubøy
"rose" subst appell fem be ent
; "rosa" adj be ent pos REMOVE:2311
<word>cupcakes</word>
"<cupcakes>"
"cupcakes" ukjent
<word>hører</word>
"<hører>"
"høre" verb pres
<word>kanskje</word>
"<kanskje>"
"kanskje" adv
<word>med</word>
"<med>"
"med" prep
<word>når</word>
"<når>"
"når" sbu SELECT:2579
; "nå" verb pres SELECT:2579
; "når" adv REMOVE:3383
<word>man</word>
"<man>"
"man" pron ent pers hum nom
 SELECT:3451
; "man" subst appell fem ub ent
 SELECT:3451
; "man" subst appell mask ub ent
 SELECT:3451
; "mane" verb imp SELECT:3451
<word>skal</word>
"<skal>"
"skulle" verb pres <aux1/perf_part>
 <aux1/infinitiv>
<word>ha</word>
"<ha>"
"ha" verb inf <aux1/perf_part>
 SELECT:2430
; "ha" interj SELECT:2430
; "ha" subst symb REMOVE:3574
; "ha" verb imp <aux1/perf_part>
SELECT:2430
<word>bloggtreff</word>
"<bloggtreff>"
"bloggtreff" subst appell nøyt ub ent
 samset-analyse <+treff>
"bloggtreff" subst appell nøyt ub fl
 samset-analyse <+treff>
<word>?</word>
"<?>"
"$?" clb <<< <spm> <<<
```

Figure B: Tokenised, multitagged and disambiguated sentence

Comparing Methods for Segmenting Elementary Discourse Units in a French Conversational Corpus

Laurent Prévot
Aix Marseille Université & CNRS
Laboratoire Parole et Langage
Aix-en-Provence, France
laurent.prevot@univ-amu.fr

Julie Hunter
LINAGORA Labs
Toulouse, France
hunter@linagora.com

Philippe Muller
Toulouse Université & CNRS
IRIT
Toulouse, France
philippe.muller@irit.fr

Abstract

While discourse segmentation and parsing has made considerable progress in recent years, discursive analysis of conversational speech remains a difficult issue. In this paper, we exploit a French data set that has been manually segmented into discourse units to compare two approaches to discourse segmentation: fine-tuning existing systems on manual segmentation vs. using hand-crafted labeling rules to develop a weakly supervised segmenter. Our results show that both approaches yield similar performance in terms of f-score while data programming requires less manual annotation work. In a second experiment we play with the amount of training data used for fine-tuning systems and show that a small amount of hand labeled data is enough to obtain good results (albeit not as good as when all available annotated data are used).

1 Introduction

Discourse parsing is the decomposition of texts or conversations in functional units that encode participants intentions and their rhetorical relationships. Segmentation in these units is the first step for other levels of analysis, and can help downstream NLP tasks.

Discourse parsing involves determining how each part of a discourse contributes to the discourse as a whole—whether it answers a question that has been asked, provides an explanation of something else that was said, or signals (dis)agreement with a claim made by another speaker. The first step, then, is to decompose a discourse into minimal parts that can serve such discursive functions. We will use the term *elementary discourse unit* (EDU; Asher and Lascarides, 2003) to designate a minimal speech act or communicative unit, where each EDU corresponds roughly to a clause-level content that denotes a single fact or event. While EDU segmentation of written documents has received a lot of attention from the discourse and NLP community, this is less true for segmentation of conversational speech. Conversational data has mostly been approached from either (i) a dialogue act segmentation and tagging perspective, and usually on rather task-oriented dialogues (Dang et al., 2020) or (ii) punctuation prediction to enrich transcripts obtained with Automatic Speech Recognition (Batista et al., 2012).

We assume that this situation encourages a bias toward written genres that can be problematic for discourse segmentation, because those genres (newspapers, literature,...) tend to include long complex sentences, while actual conversations are made of relatively short contributions. Non-sentential units (Fernández et al., 2007), which often consist of only a single word, are extremely frequent in conversation and can convey full communicative acts, such as an answer to a question or a communicative feedback, that are crucial for modeling discourse structure.

In this paper we benefit from a fully segmented corpus, the Corpus of Interactional Data (CID; Blache et al., 2017), for running a set of experiments on discourse segmentation. This data set is challenging as it consists of 8 long conversations (1 hour each) alternating between interactive narrative sequences like (1) (with a clear dominant speaker holding long turns made of many discourse units) and more dialogical sequences like (2) (made of very short, very often incomplete, turns

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 436–446

that sometimes need to be grouped together to form a valid discourse unit).

(1) [on y va avec des copains]$_{du}$ [on avait pris le ferry en Normandie]$_{du}$ [puisque j'avais un frère qui était en Normandie]$_{du}$ [on traverse]$_{du}$[1]
[we went there with friends]$_{du}$ [we took the ferry in Normandy.]$_{du}$ [since I had a brother who was in Normandy]$_{du}$ [we cross]$_{du}$

(2) A : tu vois à peut près où ///c'est///[2]
you know more or less where ///it is///
B: ///oui///
///yes///
A: ouais
yeah

EDUs have become a central topic for discourse parsing (Zeldes et al., 2019a), and in this paper we present two main experiments designed to (i) compare EDU segmentation approaches on our challenging data set and (ii) evaluate the impact of the amount of hand-labeled data used for training. More precisely we first compare the results of several baselines with (i) *state-of-the-art* level segmentation systems fine-tuned on the CID and (ii) a weakly supervised approach bootstrapped by hand crafted labeling functions. Our second experiment consists in varying the amount of hand-labeled data used either for training the base model from scratch or for fine-tuning an existing written text segmenter.

2 Previous and Related work

2.1 Discourse Segmentation

Discourse segmentation had been largely neglected by work in discourse parsing, and mostly applied to English corpora (Wang et al., 2018), until a few years ago when the multilingual, multi-framework `disrpt` campaigns were introduced (Zeldes et al., 2019b, 2021a). The present paper relies heavily on the French model, Tony (Muller et al., 2019), from those campaigns. Built on the Allen NLP library (Gardner et al., 2018), Tony is a sequential tagging model over contextual embeddings, namely multilingual BERT (Devlin et al., 2018), that treats segmentation as a token classification problem where each token is either a beginning of a segment or not.

While the overall best segmenter on the `disrpt` tasks is currently Gessler et al. (2021), this segmenter requires syntactic parsing, which is unreliable for highly spontaneous conversational speech of the kind in CID. Moreover, for the sake of our experiment, simple pipelines (based on plain text for Tony) are preferable to more sophisticated ones. Finally, Tony is on par with the best performing French model? (Bakshi and Sharma, 2021). See (Zeldes et al., 2021b) for details on the performance of existing systems.

Gravellier et al. (2021) adapted Tony to conversational data by (i) fine-tuning it on a conversational data set and (ii) adopting a data-programming approach similar to what we propose here. However, the transcriptions used in their work were obtained from a recording setting with a unique microphone. In CID, each participant is recorded on a separate channel, and the transcription of the corpus was fully manual and even corrected several times to reach very high transcription accuracy for a conversational corpus. Moreover, only a small portion of the corpus used by Gravellier et al. (2021) contains gold EDU segmentation ($\approx$ 1100 units), as this corpus was segmented to train a weakly supervised labeling model guided by hand-crafted labeling rules. The present work is grounded on a completely different data set, CID (Blache et al., 2009), that has been fully manually segmented. This corpus provides over 17000 discourse units to experiment with, which allows us to evaluate the quantity of supervised data that is needed to equal or improve performance over the weakly supervised model.

2.2 Weak supervision

For the weak supervision part of our experiments, we rely on the so-called "data programming" approach proposed by Ratner et al. (2017). The general principle is to design multiple overlapping heuristic rules for a classification problem, then aggregate them statistically to automatically produce noisy labels on unannotated data that can then be

[1] Color alternation is used to highlight discourse units.

[2] ///***/// indicates overlapping speech.

fed to a regular supervised model.

This approach has been implemented in the Snorkel library (Ratner et al., 2017) and also independently adapted in the Skweak framework of Lison et al. (2021) and the Spear library (Abhishek et al., 2022). These frameworks provide both an API to define heuristic rules and an aggregation model for the rules. Their output is a noisily annotated data set that can then be used to train the supervised model of one's choice. This approach has been used in discourse analysis to enrich a discourse parser (Badene et al., 2019), and is also the basis of the work in Gravellier et al. (2021) mentioned above. For our final supervised model, we adopted the architecture of Gravellier et al. (2021), but trained it on a different noisy data set.

As explained above, the general idea behind data programming is to leverage expert knowledge by writing a set of labeling functions (*LF*) that can be developed and tested over a very small amount of annotated development data. The system builds a profile for each (*LF*) and a model is trained by combining all *LFs* (LFs being weighted by their accuracies). This model is then used for labeling a training set and finally a supervised model is trained on the data set that had been automatically annotated by the label model.

These frameworks leave open the choice of the final supervised model, since their output is just a (noisily) annotated data set. As the final supervised model, we used the same architecture as previously mentioned work on segmentation Gravellier et al. (2021), but only train it on the noisy data set.

3 Gold EDU segmentations

The Corpus of Interactional Data (CID) (8 dyadic conversations, 1 hour duration for each) (Blache et al., 2009, 2017) was segmented following guidelines designed for written documents (Muller et al., 2012) that were modified for spoken conversational data. These guidelines thus combine semantic and discourse criteria (used in particular in monological sequences like (1)) with dialogical and interactional ones (that are more useful in dialogical sequences like (2)). The CID displays highly spontaneous data with colloquial sequences like (3) or strong disfluencies like (4) making discourse segmentation a much more difficult task than on written genres, even for humans. The whole data set consists of about 125 000 tokens for 15,463 discourse units (12,4% of the tokens are EDU boundaries). EDUs are obtained from at least two manual annotations (obtained from 4 naive coders and 2 experts). The mean Cohen's κ-score across speaker for naive coders is 0.85 (min: 0.83 ; max :0.87). Annotations were performed with Praat (Boersma, 2002) in order to have access to signal word-alignment when making segmentation decisions. The discourse annotations (Prévot et al., 2021) are available from Ortolang platform : `https://www.ortolang.fr/market/item/ortolang-000918`.

(3) A: [comme ça # ah ouais non c'était]$_{du}$
A: [like that # oh yeah no it was]$_{du}$
B: [ah ouais profitez profitez de vos soirées]$_{du}$
B: [oh yeah enjoy enjoy your evenings]$_{du}$
A: [ouais c'est pour ça]$_{du}$
A: [yeah it's for that]$_{du}$

(4) [ou des euh non pas des f- pas des frustrations]$_{du}$ [des # espèces de euh # mhm # ouais des des vues différentes sur le boulot quoi]$_{du}$
[or some uh no not some f- not some frustrations]$_{du}$ [some kind of uh # mh # yeah some some different views about work what]$_{du}$

4 Method

In this work we use the existing implementation of Tony (Muller et al., 2019) and that of Gravellier et al. (2021), called `tony-w(ritten)` and `tony-s(poken)`, respectively, as baselines. Our first experiment consists in comparing a supervised model (fine-tuning 'Tony' baselines with our annotated data) against the weakly supervised data-programming approach. In a second experiment, we explore the impact of the amount of data used for fine-tuning the models.

Transcripts from the CID do not include any

name	Polarity	Coverage	Overlaps	Conflict	Correct	Incorrect	Accuracy
tony_written	0,1	1.000	0.976	0.0632	102446	8994	0.919
no_pause	0	0.898	0.898	0.0474	94706	5415	0.946
long_pause	1	0.054	0.054	0.0094	5353	683	0.887
very_long_pause	1	0.032	0.032	0.0051	3399	162	0.955
extreme_pause	1	0.022	0.022	0.0037	2357	48	0.980
pause_begin_pos	1	0.042	0.042	0.0101	4362	328	0.930
pause_ending_pos	1	0.036	0.036	0.0097	3338	652	0.837
non_ending_tok	0	0.323	0.323	0.0009	35579	365	0.990
pause_begin_tok	1	0.055	0.055	0.0111	5533	617	0.900
pause_ending_tok	1	0.005	0.005	0.0005	550	22	0.962
dm_bi_ini	1	0.012	0.012	0.0059	1146	222	0.838
non_begin_tok	0	0.005	0.005	0.0001	530	9	0.983
feedback_cluster	0	0.026	0.026	0.0012	2763	87	0.969
repeat	0	0.088	0.088	0.0124	8678	1141	0.884
filled_pause	0	0.071	0.071	0.0055	7428	570	0.929
truncated_word	0	0.016	0.016	0.0013	1629	169	0.906

Table 1: Labeling Function statistics

kind of punctuation ((punctuating conversational speech was taken to be a complex pragmatic annotation task that relies on prosody and other sources of information that are not part of the transcription process). Punctuation, however, is a crucial cue for existing discourse segmenters based on written text. We therefore decided to introduce breaks by treating all pauses in our experiments that were over 200 ms as introducing commas in the token sequence, and all pauses over 900 ms as indicators of "document separation" (like a period in written text). This allowed to help the baseline models trained on textual data and written genres. The idea behind such a short (200 ms) pause duration is that pauses signal places in which a discourse segmentation is likely to happen. When facing these pause/comma tokens, the systems then try to distinguish those corresponding to discourse breaks from the other ones. This does not mean that the system does not predict discourse boundaries at other locations.

4.1 Fine Tuning

Fine-tuning of both `tony-w` and `tony-s`– where the latter results from fine-tuning the former with data from a conversational corpus using the data programming approach– proceeded in the same fashion. We first continued to train the original models with the same configurations but with CID labeled data. We conducted a cross-validation experiment in which 7 conversations (7 hours) are used for fine-tuning both models and tested on the remaining eighth conversation, and performed a permutation to obtain a cross-validation.

4.2 Data-Programming

Like Gravellier et al. (2021), we pulled from our knowledge of conversational French to define hand-crafted rules (i.e. labeling functions) for the data programming approach. Our approach differed in important ways from that of Gravellier et al. (2021), however, stemming in part from difference in the target data sets and also preprocessing choices. While (Gravellier et al., 2021) attempted to exploit more prosodic and acoustic information in their labeling functions, our rules are based solely on time-aligned (at token level) transcription, NLP annotations (POS-tagging) and duration (in particular pause duration). We also opted for a different POS-tagger: while (Gravellier et al., 2021) used Spacy (Honnibal and Montani, 2017) we chose Stanza (Qi et al., 2020) because it offers a 'spoken' model for French which proved to be more reliable than Spacy for tagging crucial tokens specific to conversational speech. Both the original Tony model and the model developed by Gravellier et al are used to define heuristic LFs.

Table 1 presents the most important labeling functions (LF) retained after various experiments on the development set. The columns of this table are the ones produced by the *La-*

```
@labeling_function()
def long_pause(x):
    return BEG if (x["prev-tok"]=='#' and x['prev_dur'] > LONG_PAUSE) else ABSTAIN

@labeling_function()
def non_ending_tok(x):
    return NOBEG if x["prev-tok"] in NON_ENDING else ABSTAIN

@labeling_function()
def pause_and_ending_pos(x):
    if ((x["prev-tok"]=='#') and (x["prev_dur"] > PAUSE)
            and (x["pprev-pos"] in ENDING_POS)):
        return BEG
    else:
        return ABSTAIN

@labeling_function()
def repeat(x):
    return NOBEG if x["tok"] in [x["prev-tok"],x["pprev-tok"],x['ppprev-tok']]
                                                  else ABSTAIN

@labeling_function()
def filled_pause(x):
    return NOBEG if ((x["prev-tok"] in FP) or (x["tok"] in FP))  else ABSTAIN

@labeling_function()
def truncated_word(x):
    if (str(x["prev-tok"])[-1]=='-'):
        return NOBEG
    elif ((str(x["pprev-tok"])[-1]=='-') and (x["prev-tok"] in [',','*','euh'])):
        return NOBEG
    else:
        return ABSTAIN
```

Figure 1: Labeling Function examples

beling Function Analysis function provided by Snorkel: *Polarity* states whether the LF labels a boundary or the absence of a boundary; *Coverage* corresponds to the percentage of instances for which the LF was triggered; *Overlaps* quantifies the proportion of times other LFs are firing at same time as a given LF; *Conflict* quantifies whether any other LFs predict a different label; *Correct/Incorrect* is the amount of correct/incorrect labels based on the development data set and this also defines Accuracy.

Unsurprisingly, `tony` contributes significantly to the prediction of segment boundaries. Rules based on pause duration (e.g. `long_pause` in Figure 1) and POS also had a considerable impact on results as did lists of tokens associated with the presence or absence of EDU boundaries. NON-ENDING TOKENS, for example, include various pronouns, determiners, prepositions, negations and initiating discourse markers (See full list in the Appendix). Most of the selected rules involving tokens and POS use pauses as additional criteria (`pause_ending_pos`). The rules `repeat`, `filled_pause` and `truncated_word` target disfluencies, which are generally associated with the absence of an EDU bound-

ary. Finally `feedback_cluster` targets sequences of acknowledgement tokens that generally constitute a single EDU *(e.g., ah ouais d'accord/oh yeah right)*.

The main interest of data-programming is to aggregate sources of information to perform the classification task. For our purposes, the core idea was to combine text-based existing segmentation models with conversational/spoken expert knowledge expressed via labeling functions, and thus our discussion in Section 5 focuses on results the include `tony`. We note, however, that it is possible to compare the results with data-programming models that do not rely on an existing text-based segmentation model. These models tend to have much higher precision (> 0.75) but low recall (< 0.6) and overall, a lower f-score (≈ 0.67). This is due to the fact that the expert LFs are rather precise but fail to cover many cases common to monological sequences in conversation and monologue in text, where `tony` excels. On the other hand, `tony` tends to predict too many boundaries, leading to the drop in precision observed when its predictions are taken into account.

4.3 Amount of labeled data

We experimented on varying the amount of labeled data used to train the supervised model (10, 20, 30, 50 or 80%). This was done either as fine-tuning of `tony-w (ft)` or as direct training from the base model (`no-model`) which is a simple BERT model in our case.

5 Results and Discussion

5.1 Fine-tuning vs. Data programming

The results of the baselines, fine-tuning and the data-programming approach constitute a global coherent picture. Tony baselines show a high recall (Figure 2) but with a relatively low precision (Figure 3). `tony-spoken` starts out significantly worse than `tony-written`. This is probably due to (i) the relative low quality of the transcriptions used for training `tony-spoken` and perhaps the nature of our data which is conversational but hosts a significant amount of narrative sequences.

The baselines based on pause duration only (we show here pause baselines at 200ms and

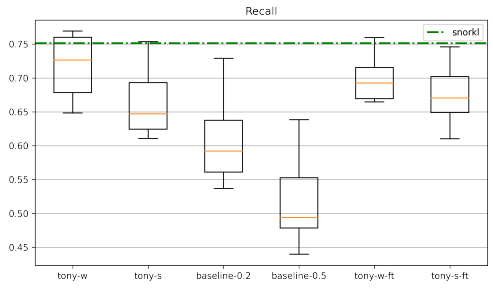


Figure 2: Boundary recall for various configurations; green dashed line = Data-programming

500ms) exhibit a surprisingly high precision, showing the relevance of using this cue as a signal for discourse units. They do miss quite a few cases but overall perform well (especially with a threshold of 200ms). The missing boundaries are discourse units not separated by any pauses, like in (1) for example.

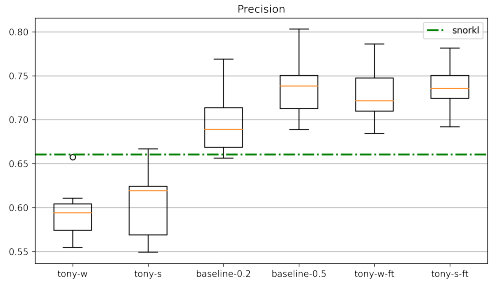


Figure 3: Boundary Precision for various configurations, green dashed line = Data-programming

Fine-tuning really helps `tony` models: recall remains high and precision increases significantly (Figure 4). Fine-tuning allows the model to distinguish which commas (pauses) do not introduce discourse segments.

The comparison of f-scores (Figure 4) of fine-tuning and data-programming approaches does not yield significant differences. It seems to validate the interest of the weakly supervised data programming approach since writing the labeling rules requires much less effort than manually segmenting a large corpus.

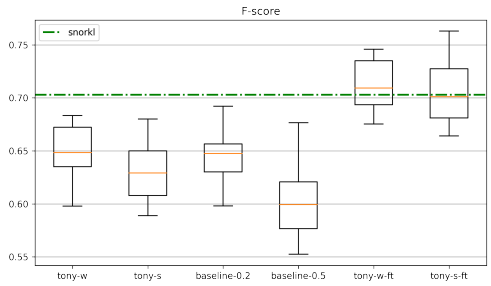


Figure 4: Boundary F-score; green dashed line = Data-programming

5.2 Amount of hand labeled data

In our second experiment, we incrementally reduced the amount of annotated data used to train a supervised model in order to determine whether performance would decrease strongly if only a small amount of annotated data were provided.

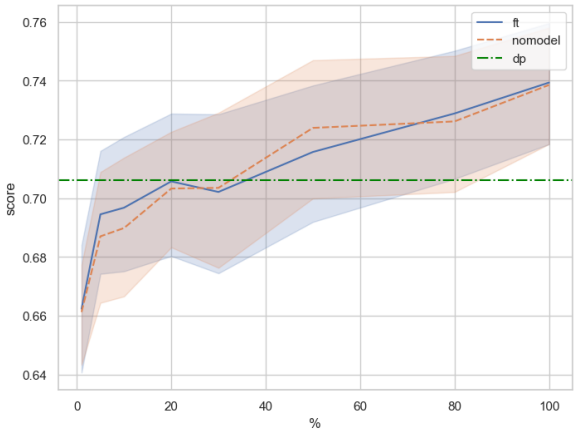


Figure 5: F-score for the supervised models trained on 1, 5, 10, 20, 30, 50, 80% of the original training data set (nomodel: just base BERT, ft: fine-tuned). We also indicate the score of the data programming model (dp). Bands are 95% confidence interval based on the cross-validation.

The results presented in Figure 5 suggest that even if more data is better, a small amount of training data (here 10% corresponds to about 1500 discourse units which is still a significant annotation effort) is enough to really improve the base model. This result mitigates the previous finding: since efficiently fine-tuning existing models does not seem to require annotating a lot of data, the difference in terms of efficiency between hand-labeling and developing a set of labeling functions is not huge, suggesting that both approaches are worth exploring depending on the use case.

5.3 Error Analysis

Error analysis of both data programming and the fine-tuned models yields interesting observations. As expected, phenomena that are really specific to conversational speech are the main sources of errors. For example *'quoi'/'what'* is a very common French function word that is heavily used in final position in conversational speech (with a rather unclear function) (Delafontaine, 2020). This item was a major source of error.

Relative clauses anchored on extremely light hosts were also problematic, particularly when they had a restrictive function as in (5). The data-programming approach tended to segment after relative pronouns whether they introduced restrictive or non-restrictive clauses, which generated some over-segmentation.

(5) [genre des gens || qui étaient au même niveau que moi]$_{du}$
[like people || that were at the same level as me]$_{du}$[3]

Another important source of error was complex disfluencies involving discourse markers as illustrated in (6).

(6) [mais là # || mais euh || mais là c' est normal]$_{du}$
[but in that case ||# but uh || but in that case it is normal]$_{du}$

The fine-tuned model introduced errors of its own. It did not segment on certain discourse marker cues like *'mais'/'but'* and *'si'/'if'*. It does not seem to judge them to be reliable initiators of discourse units.

A second source of error for this model was the repetition of presentative constructions *'c'est'/'it is'* of which an extreme example is given in (7).

[3]In the error analysis examples, || stands for a false positive (added a boundary in a wrong place) and $$ for false negative (missed a boundary).

(7) [non c' est plus de la recherche]$_{du}$ $$ [c' est de la c' est de la # c' est # a-]$_{du}$ # [ouais voilà]$_{du}$ # [c' est de la t- c- c' est]$_{du}$ $$ # [co- comment ça s' appelle]$_{du}$ $$ [c' est de la # || de la capitalisation]$_{du}$

[no it is not research anymore]$_{du}$ $$ [it is some it is some # it is # a-]$_{du}$ # [yeah that's right]$_{du}$ [# it is some t- c- it is h-]$_{du}$ # [how do you call it]$_{du}$ $$ [it is some # some capitalisation]$_{du}$

There are a wide range of other errors represented, though they are less frequent. They include (i) long pauses (>1s) that do not actually split a discourse unit. As explained above, our preprocessing step splits 'documents' based on pauses that last more than 900ms, and while during fine-tuning, the models see that a 'document start' does not always correspond to a 'discourse unit start', document starts tend to be used for detecting boundaries (because they are always in the model before fine-tuning); (ii) dialogical structures (involving both speakers) that are currently not handled; (iii) reported speech (that was systemically segmented in the manual annotation even if sometimes the reported speech introduction was extremely light in content).

In the CID, there are two kinds of sequences: (i) narrative sequences in which one of the participants tells a story (with an interactive flavor involving feedback and production from the other participant but in which there is a clear main speaker and a narrative flow), and (ii) transition sequences where the participants comment and chat about these stories, as well as negotiate who will tell the next story and what it will be about. As expected, narrative sequences are better handled by our models, even when produced at a relatively fast pace that did not allow for pauses between discourse units like (8) which is the continuation of our example (1) and in which there are no pauses (longer than 200 ms) but several discourse units.

(8) [on y va avec des copains]$_{du}$ [on avait pris le ferry en Normandie]$_{du}$ [puisque j' avais un frère qui était en Normandie]$_{du}$ [on traverse]$_{du}$ [on avait passé une nuit épouvantable sur le ferry]$_{du}$

–

[we went there with friends]$_{du}$ [we took the ferry in Normandy.]$_{du}$ [since I had a brother who was in Normandy]$_{du}$ [we cross]$_{du}$ [we spent a terrible night on the ferry]$_{du}$

However, even in narrative sequences some common spoken constructions seem to cause problems for the models, including presentatives such as *y a/y avait* (Lambrecht, 1988) in (9).

(9) [on est rentré dans un bar # qui faisait boîte]$_{du}$ $$ [y avait # que nous]$_{du}$ # $$ [y avait la musique # à fond les ballons]$_{du}$

–

[we entered a bar # that was also a nightclub]$_{du}$ $$ [there was # only us]$_{du}$ # $$ [there was music # (that was) extremely loud]$_{du}$

6 Conclusion and Discussion

In this work we have compared different approaches for building a discourse unit segmenter adapted to French conversations. We had access to a manually segmented corpus of significant size which allowed us to perform a wide range of experiments. First we compared the option of (i) using our conversational data set to fine-tune an existing discourse segmenter developed and trained for written data, (ii) a data-programming approach that makes use of the same "text-based" discourse segmenter but enriched with manual defined rules (Labeling Functions). We found that both approaches yielded similar results. This suggests that both approaches are worth considering depending on the exact use case. While data-programming requires some heuristic rule engineering, fine-tuning requires annotated data that is costly to obtain, especially for relatively expert tasks such as discourse segmentation.

We also ran a second experiment to investigate (i) how much manually annotated data is required before reaching the same performance as the data-programming approach; (ii) whether starting from a written base segmentation model was useful at all (compared to

training the segmenter directly over the BERT pretrained language model). To the first question, it is noteable that given the significant variability between folds, only a small amount of annotated data ($\sim$ 20%) is sufficient to get close to the best results we obtained. Moreover, starting from a written discourse segmenter model or directly from BERT did not significantly change the results.

Overall, our findings suggest that annotating (segmenting) a large amount of conversation might not be necessary since both the data-programming approach (that makes use of an existing discourse segmenter developed on written data) and a model trained with little data (here about 2700 discourse units) yielded results comparable to a model finedtuned on our whole training set (13500 discourse units).

Breaking down the performance of the different models, we see that both the fine-tuned model and the weakly supervised one improve over the pause baselines. While pauses are strong predictors (rather high precision for a baseline), many discourse units are not preceded by a pause, so extra cues are needed. Both models seem to easily learn how to segment within "fluent monologues" (even without pauses)—an result likely explained at least in part by the role of the existing discourse segmenter and the relevant language model. However, when speech becomes strongly disfluent,[4], in particular when disfluency gets tangled up with discourse markers that typically signal discourse segment starts, both approaches struggle. Finally, certain constructions, such as presentatives, which are frequent in conversational language but absent from written data, are also an issue. Overall while the models are definitely useful as discourse segmenters, their scores are way below the state-of-the-art obtained on written text. Apart from the fact that the task of EDU segmentation is arguably more difficult for spoken language, underlying biases carried by segmenters trained on written data could explain in part why our models remain relatively confused when facing token sequences found only in conversational data sets, despite fine-tuning or our attempt to add heuristic specific rules.

As future work, we plan to refine our experiments by separating discourse units into two categories: easy and difficult to segment. Indeed, in conversation, a sizeable amount (about 19%) of discourse units are trivial to segment (single or lexical feedback items preceded and followed by long pauses) while some others, as we have seen in error analysis, are really complex to delineate. Our opinion is that separating these two cases at all stages (including inter-coder agreement measures) will allow us to learn more about discourse segmentation of conversation and ultimately help in developing better performing models.

[4]Both approaches seem to overcome simple disfluency, like the presence a filled pause and/or a content word repetition, relatively well.

Ethics Statement

This paper does not process any sensitive material and does not generate any content. It does not raise any ethical issues.

References

Guttu Abhishek, Harshad Ingole, Parth Laturia, Vineeth Dorna, Ayush Maheshwari, Ganesh Ramakrishnan, and Rishabh Iyer. 2022. SPEAR : Semi-supervised data programming in python. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 121–127, Abu Dhabi, UAE. Association for Computational Linguistics.

Nicholas Asher and Alex Lascarides. 2003. *Logics of conversation*. Cambridge University Press.

Sonia Badene, Kate Thompson, Jean-Pierre Lorré, and Nicholas Asher. 2019. Weak supervision for learning discourse structure. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2296–2305, Hong Kong, China. Association for Computational Linguistics.

Sahil Bakshi and Dipti Sharma. 2021. A transformer based approach towards identification of discourse unit segments and connectives. In *Proceedings of the 2nd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2021)*, pages 13–21, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fernando Batista, Helena Moniz, Isabel Trancoso, and Nuno Mamede. 2012. Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts.

IEEE transactions on audio, speech, and language processing, 20(2):474–485.

Philippe Blache, Roxane Bertrand, and Gaëlle Ferré. 2009. Creating and exploiting multimodal annotated corpora: the toma project. *Multimodal corpora*, pages 38–53.

Philippe Blache, Roxane Bertrand, Gaëlle Ferré, Berthille Pallaud, Laurent Prévot, and Stéphane Rauzy. 2017. The corpus of interactional data: A large multimodal annotated resource. In *Handbook of Linguistic Annotation*, pages 1323–1356. Springer.

P. Boersma. 2002. Praat, a system for doing phonetics by computer. *Glot international*, 5(9/10):341–345.

Viet-Trung Dang, Tianyu Zhao, Sei Ueno, Hirofumi Inaguma, and Tatsuya Kawahara. 2020. End-to-End Speech-to-Dialog-Act Recognition. In *Proc. Interspeech 2020*, pages 3910–3914.

François Delafontaine. 2020. Unités grammaticales et particule discursive «quoi». *Studia linguistica romanica*, (4):74–91.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Raquel Fernández, Jonathan Ginzburg, and Shalom Lappin. 2007. Classifying non-sentential utterances in dialogue: A machine learning approach. *Computational Linguistics*, 33(3):397–427.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*.

Luke Gessler, Shabnam Behzad, Yang Janet Liu, Siyao Peng, Yilun Zhu, and Amir Zeldes. 2021. DisCoDisCo at the DISRPT2021 shared task: A system for discourse segmentation, classification, and connective detection. In *Proceedings of the 2nd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2021)*, pages 51–62, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lila Gravellier, Julie Hunter, Philippe Muller, Thomas Pellegrini, and Isabelle Ferrané. 2021. Weakly supervised discourse segmentation for multiparty oral conversations. In *Proceedings of EMNLP 2021*.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.

K. Lambrecht. 1988. Presentational cleft constructions in spoken French. *Clause combining in grammar and discourse*, pages 135–179.

Pierre Lison, Jeremy Barnes, and Aliaksandr Hubin. 2021. skweak: Weak supervision made easy for nlp. *arXiv preprint arXiv:2104.09683*.

Philippe Muller, Chloé Braud, and Mathieu Morey. 2019. Tony: Contextual embeddings for accurate multilingual discourse segmentation of full documents. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 115–124. Association for Computational Linguistics.

Philippe Muller, Marianne Vergez-Couret, Laurent Prévot, Nicholas Asher, Benamara Farah, Myriam Bras, Anne Le Draoulec, and Laure Vieu. 2012. Manuel d'annotation en relations de discours du projet annodis. Technical Report 21, CLLE-ERS, Toulouse University.

Laurent Prévot, Roxane Bertrand, and Stéphane Rauzy. 2021. Investigating disfluencies contribution to discourse-prosody mismatches in french conversations. In *The 10th Workshop on Disfluency in Spontaneous Speech*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access.

Yizhong Wang, Sujian Li, and Jingfeng Yang. 2018. Toward fast and accurate neural discourse segmentation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 962–967, Brussels, Belgium. Association for Computational Linguistics.

Amir Zeldes, Debopam Das, Erick Galani Maziero, Juliano Antonio, and Mikel Iruskieta. 2019a. The disrpt 2019 shared task on elementary discourse unit segmentation and connective detection. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 97–104.

Amir Zeldes, Debopam Das, Erick Galani Maziero, Juliano Antonio, and Mikel Iruskieta. 2019b. The DISRPT 2019 shared task on elementary discourse unit segmentation and connective detection. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*,

pages 97–104, Minneapolis, MN. Association for Computational Linguistics.

Amir Zeldes, Yang Janet Liu, Mikel Iruskieta, Philippe Muller, Chloé Braud, and Sonia Badene. 2021a. The DISRPT 2021 shared task on elementary discourse unit segmenta tion, connective detection, and relation classification. In *Proceedings of the 2nd Shared Task on Discourse Relation Parsin g and Treebanking (DISRPT 2021)*, pages 1–12, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Amir Zeldes, Yang Janet Liu, Mikel Iruskieta, Philippe Muller, Chloé Braud, and Sonia Badene, editors. 2021b. *Proceedings of the 2nd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2021)*. Association for Computational Linguistics, Punta Cana, Dominican Republic.

A Useful token list for Labelling Functions

- BEGIN_TOK = DM_INI + PRO_SUJ + FEEDBACK
- ENDING_TOK = quoi,hein
- NON_ENDING_TOK = PRO_SUB + PRO_OTH + PRO_REL + DEM + DET + PREP + OTHER + NEG + DM_INI

with the following lists:

- PRO_SUB = je, tu, il, vous, on, nous, elle, ils, elles, j',c',t',y
- PRO_OTH = me, te, se, mes, tes, ses, mon, ton, son, ma, ta, sa, nos, vos , leur ,ceux
- PRO_REL = qu', que, qui, quel
- DEM = ce, cette, ces, cet
- DET = le, la, les, un, une, des , l', d'
- PREP = à, de, par, pour, en, dans, chez, sur, sous, pendant, avec
- OTHER = soit, juste, pendant, surtout, chaque, quelque, quelques, sauf
- NEG = n', ne
- DM_INI = mais, donc, parce, ah, alors, c'est-à-dire, puisque, bah
- FEEDBACK = mh, ouais, ah, oui, bon, voilà, putain,oh, okay,ok,euh,ben,et,d'accord, non

B Part-of-Speech list for Labelling Functions

- NON_ENDING_POS = DET, CCONJ, SCONJ, ADP
- ENDING_POS = INTJ
- BEGINNING_POS = INTJ, CCONJ, SCONJ
- NON_BEGINNING_POS = VERB, AUX
- NO_RULES = ADJ, NOUN, ADV, NUM, PROPN, X

Multi-way Variational NMT for UGC: Improving Robustness in Zero-shot Scenarios via Mixture Density Networks

José Carlos Rosales Núñez
Université Paris Saclay & LISN
INRIA Paris
jose.rosales-nunez@inria.fr

Djamé Seddah
INRIA Paris
djame.seddah@inria.fr

Guillaume Wisniewski
Université Paris Cité,
LLF, CNRS
guillaume.wisniewski@u-paris.fr

Abstract

This work presents a novel Variational Neural Machine Translation (VNMT) architecture with enhanced robustness properties, which we investigate through a detailed case-study addressing noisy French user-generated content (UGC) translation to English. We show that the proposed model, with results comparable or superior to state-of-the-art VNMT, improves performance over UGC translation in a zero-shot evaluation scenario while keeping optimal translation scores on in-domain test sets. We elaborate on such results by visualizing and explaining how neural learning representations behave when processing UGC noise. In addition, we show that VNMT enforces robustness to the learned embeddings, which can be later used for robust transfer learning approaches.

1 Introduction

The specificities of user-generated content (UGC) leads to a wide range of vocabulary and grammar variations (Foster, 2010; Seddah et al., 2012; Eisenstein, 2013). These variations result in a large increase of out-of-vocabulary words (OOVs) in UGC corpora with respect to canonical parallel training data and raise many challenges for Machine Translation (MT), all the more since common language variations found in UGC are actually productive (there will always be new forms that will not have been seen during training). This fact limits the pertinence of "standard" domain adaptation methods such as fine-tuning[1] or normalization techniques (Martínez Alonso et al., 2016) and urges the development of robust machine translation models able to cope with out-of-distribution (OOD) texts in a challenging zero-shot scenario in which the target distribution is unknown during training.

To address the problem raised by OOD texts, an increasing number of works (Setiawan et al., 2020; Schmunk et al., 2013; McCarthy et al., 2020; Przystupa, 2020; Xiao et al., 2020) explore the possibility to combine deep learning with latent variable (LV) models, which are indeed able to capture underlying structure information and to model unobserved phenomena. The combination of these models with neural networks was shown to increase performance in several NLP tasks (Kim et al., 2018). In this work, we focus on a specific latent variable model for MT, Variational NMT (VNMT) (Zhang et al., 2016) which has been reported to have good performance and interesting adaptability properties (Przystupa, 2020; Xiao et al., 2020).

The goal of this work is twofold. First, we aim to evaluate the performance of VNMT when translating a special kind of OOD texts: French social-media noisy UGC. To account for the challenges raised by the productive nature of UGC, we consider a highly challenging zero-shot scenario and assume that only canonical texts[2] are available for training the system. We hypothesize that, by leveraging on Variational NMT, latent models can build more robust representations able to represent OOD observations that are symptomatic of noisy UGC and automatically map them to in-distribution instances, which can be more easily translated.

Furthermore, to account for the diversity of UGC phenomena, we introduce a new extension of VNMT that relies on Mixture Density Networks (Bishop, 1994) and Normalizing Flows (Rezende and Mohamed, 2015). Intuitively,

[1] As the fine-tuning data will only reflect a frozen state of idiosyncrasies.

[2] We consider the corpora generally used to train MT systems as "canonical" as they contain texts following the set of standard grammatical and morphological source-language rules.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 447–459

each mixture component extracts an independent latent space to represent the source sentence and can potentially model different UGC specificities. Interestingly, extracting embeddings from our zero-shot model that has never seen any UGC data and using them in a classic transformer-based NMT model leads to a stronger, more robust to UGC noise model. This is in line with the regularizing character of VNMT (Zhang et al., 2016).

Our contributions can be summarized as follows:

- we study the performance, in a zero-shot scenario, of VNMT models and evaluate their capacity to translate French UGC into English, which resulted in a consistent improvement of translation quality;
- we introduce a new model that uses state-of-the-art transformer as the backbone of a variational inference network to produce robust representation of noisy source sentences, and whose results outperform strong VNMT and non-latent baselines when translating UGC in a zero-shot scenario. Specifically, our model demonstrates a high robustness to noise while not impacting in-domain translation performance;
- by probing the learned latent representations, we show the importance of using several latent distributions to model UGC and the positive impact of the ability of VNMT models to discriminate between noisy and regular sentences while maintaining their representation closer in the embedding space;
- we report evidence that our VNMT models act as regularizers of their backbone models, leading to more robust source embeddings that can be later transferred with a relatively high performance gain in our zero-shot UCG translation scenario.

2 Background and related works

Variational Neural Machine Translation Variational Inference (VI) methods (Kingma and Ba, 2015) are generative architectures capable, from a distributional perspective, of modeling the hidden structures that can be found in a corpus. In a sequence-to-sequence MT task, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are respectively the source and target sentences, VNMT (Zhang et al., 2016) architectures assume there exists an hidden variable $\boldsymbol{z}$ modeling the implicit structure (i.e. relations) between the bilingual sentence pairs. In the context of UGC translation, we believe that this latent variable can capture the variations between a source sentence and its canonical, normalized form, recovering its underlying meaning and ensuring that the representation of the former is close to the representation of the latter.

To make computations tractable, in spite of the latent variable, VI combines a so-called *variational posterior* $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})$ that is chosen to approximate the *true* posterior distribution, with prior $p(\boldsymbol{z}|\boldsymbol{x})$; and a neural decoder generative distribution, $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})$, in charge of generating the translation hypothesis conditioned on the latent variable. Once the family of densities q is chosen, the parameters of the two distributions are jointly estimated to model the output $\boldsymbol{y}$ by looking for the parameters $(\boldsymbol{\theta},\boldsymbol{\phi})$ that maximizes the *evidence lower bound* objective function:

$$\begin{aligned}\log p_\theta(\boldsymbol{y}) \geq \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[\log p_\theta(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})] \\ - D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})||p(\boldsymbol{z}|\boldsymbol{x})]\end{aligned} \quad (1)$$

Normalizing Flows One of the major caveats of variational methods is that choosing the prior $q(\boldsymbol{z})$ is a complicated process that requires some *a priori* knowledge of the task. In practice, a normal distribution with fixed parameters (generally $\mu = 0.0$ and $\sigma = 1.0$) is often chosen due to the simplicity of its re-parametrization for sampling. However, such an assumption can be restrictive when modeling more complex processes.

Regarding this issue, Rezende and Mohamed (2015) propose to enhance variational methods with Normalizing Flows (NF) (Tabak and Turner, 2013). A chain of normalizing flows is a series of simple bijective functions automatically chosen to extract a more suitable representation for the task at hand from a random variable, by alleviating the restrictions of choosing a default fixed prior. Concretely, a base distribution $q_0(\boldsymbol{z}_0)$, that generates the initial latent codes, $\boldsymbol{z}_0$, undergoes a series of invertible and smooth transformations $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$, called *flows*. Then, the random latent variables $\boldsymbol{z}$ are transformed to the random variable $\boldsymbol{z}' = f(z)$ after each flow:

$$q(\boldsymbol{z}') = q(\boldsymbol{z})\left|det\frac{\partial f^{-1}}{\partial \boldsymbol{z}'}\right| = q(\boldsymbol{z})\left|det\frac{\partial f}{\partial \boldsymbol{z}}\right|^{-1} \quad (2)$$

Finally, we can build an arbitrarily K-long chain of f_k transformations to generate the final latent variables, $\boldsymbol{z}_K$, from the initial random variables, $\boldsymbol{z}_0$, which is drawn from the base distribution $q_0(\boldsymbol{z}_0)$ (often chosen to be $\mathcal{N}(0,1)$):

$$\boldsymbol{z}_K = f_K \circ \ ... \circ f_2 \circ f_1(\boldsymbol{z}_0)$$
$$ln(q_K(\boldsymbol{z}_K) = ln(q_0(\boldsymbol{z_0})) - \sum_{k=1}^{K} ln\left|det\frac{\partial f_k}{\partial \boldsymbol{z}_{k-1}}\right| \tag{3}$$

In MT, normalizing flows were recently used to improve VNMT models: Setiawan et al. (2020) show that using them in an in-domain evaluation setting results in an increase of +1.3 BLEU points on the IWSLT'14 (De-En) and +0.2 BLEU points on the WMT'18 (En-De); in a *simulated* out-domain evaluation, NF still improve translation quality: adding NF to the model trained on WMT'18 result in a +0.9 BLEU score improvements than the baseline Transformer system and +0.6 compared to the VNMT without using NF.

Mixture Density Networks Mixture Density Networks (MDN) are another interesting generalization of variational encoding for modeling UGC. By using MDN, the posterior distribution of the current decoding step $p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}_t)$ is no longer approximated by a single variational distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$ but by a linear combination of variational posteriors $\tilde{q}_{\phi}^{m}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$:

$$p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}_t) = \sum_{m=1}^{M} \alpha_m(\boldsymbol{x}, \boldsymbol{y}_{1:t-1}) \cdot \tilde{q}_m(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}_{1:t-1}) \tag{4}$$

where α_m are known as the mixing coefficients. Intuitively, an MDN can be interpreted as a combination of M variational encoders. Our intuition is that, since UGC contains a large number of different kind of variations, covering very different aspects ranging from morphology to phonetics, including lexicon and sentence structure (Seddah et al., 2012); by using several independent VI components we can account for multiple UGC phenomena. Thus, with an MDN, it is possible that each component of the variational encoder is able to model different UGC specificities, allowing us to better process UGC as a whole. In the past, MDN has been used to address sequence-to-sequence generative tasks, such as `SketchRNN` (Ha and Eck, 2018) and modeling of sequential environment states in reinforcement learning (Ha and Schmidhuber, 2018).

Gumbel-Softmax sampling Regarding the mixing coefficients definition, we also explore the use of a categorical probability distribution, for which probabilities are calculated by the network, such as in Ha and Eck (2018). Unlike theirs, our supervised end-to-end training requires backpropagating the error gradient through the variational network via reparametrized sampling (Kingma and Welling, 2014) which poses optimization challenges because of the discrete random variables used as latent vector for categorical distributions. For this reason, we use the reparametrization of this distribution via the Gumbel-Softmax sampling (Jang et al., 2017; Maddison et al., 2017), such that, the $\arg\max$ function is approximated by a $\mathrm{softmax}$ and generates the relaxed one-hot encoded samples, which correspond to the mixing coefficients:

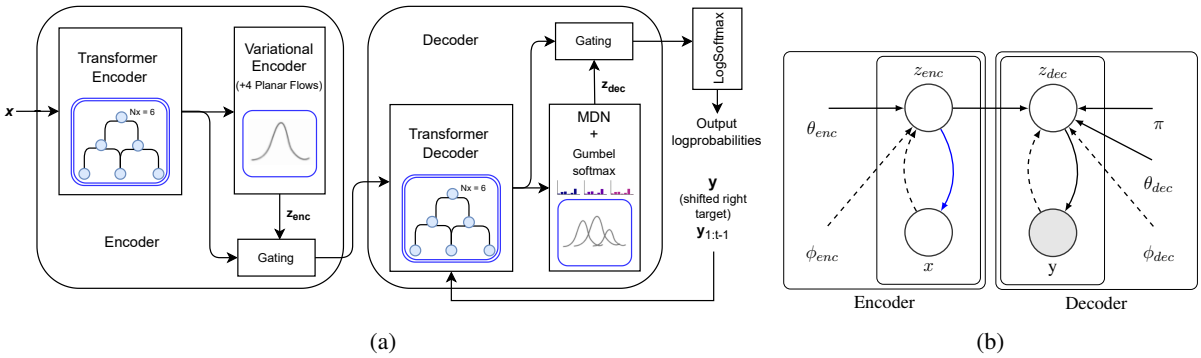
$$\alpha_m = \frac{\exp(\log(\pi_m) + g_m)/\tau)}{\sum_{j=1}^{M} \exp((\log(\pi_j) + g_j)/\tau)} \tag{5}$$

where $g_m ... g_M$ are *i.i.d* samples from the Gumbel(0,1) distribution (Gumbel, 1954; Maddison et al., 2017), π_i the probability associated to the *m*-th MDN's gaussian components, jointly generated by neural networks along with the computations of the corresponding parameters (μ_m, σ_m) for $m...M$; and τ the temperature parameter, which controls variability of the sampling. When $\tau \to 0$, the sampling exhibits a perfectly one-hot encoded output, whereas, conversely, when $\tau \to \inf$, the distribution approaches an uniform one across all the MDN's components.

3 Extending Variational Methods for Robust MT

Our model adopts a variational encoder-decoder architecture inspired by `SketchRNN` (§2) that uses an MDN on the decoder's variational network to model multiple and independent continuous generative variational distributions. However, unlike `SketchRNN`, we use a Transformer backbone for the encoder and the decoder and train our model in a end-to-end manner on canonical parallel corpora. In the following, we will first describe the general architecture of our model, denoted `multi-VNMT`, and then detail the encoder and decoder parameters.

General architecture Figure 1a represents the architecture of our model. The input sentence is



(a) (b)

Figure 1: (a) VNMT-MDN architecture overview. (b) Directed graph of our encoder-decoder model variational inference. Dashed lines represent the variational approximation for the posterior distribution, and solid lines stand for the generative models. The blue arrow depicts the generative networks for source-side monolingual reconstruction distribution $p(\boldsymbol{x}|\boldsymbol{z})$.

first processed by a standard Transformer encoder, the output of which is used by a Variational Encoder enhanced with NF to predict a latent representation of the input sentence. The latent representation and the output of the last layer of the Transformer encoder are combined using the gating mechanism of Setiawan et al. (2020).

This combined representation is then fed to the decoder that has a similar architecture: it is made of an "usual" Transformer decoder and a variational MDN that is sampled to obtain a prediction that will be combined to the Transformer output by (again) a gating mechanism.

The model can be trained in an end-to-end fashion using the "reparametrization trick" of Kingma and Welling (2014). In order to ensure that the estimated variances for the variational posteriors are positive, we used the *softplus* activation function (Zheng et al., 2015), as done in van den Berg et al. (2018)'s implementation.

In addition, concerning the training of the decoder's MDN, we compare two different ways to compute the mixing coefficient: the first one consists in a vanilla non-latent softmax, the second on a relaxed categorical variational method that relies on a Gumbel-Softmax sampling (§2).

The model has been implemented in OpenNMT-py[3] (Klein et al., 2018).

Encoder Our encoder backbone is the "standard" transformer of Vaswani et al. (2017), made of 6-layered transformer layers each with 8 attention heads. the *feed-forward* layers have 2,048 parameters and the dimension of lexical embeddings is 512. The dimension of the encoder variational network is 128. The network is extented with 4-flows Normalizing Planar Flows (Rezende and Mohamed, 2015).[4]

Following to Setiawan et al. (2020), we combine the last Transformer layer output to the latent vectors using a gating mechanism. We used a feed-forward network to transform the representation of dimension 128 predicted by the variational network into a representation of size 512 that matches the Transformer representation dimension.

In Figure 1b, we show the Transformer and variational encoding latent state $\boldsymbol{z}$ as being estimated ($p_\theta(\boldsymbol{z}|\boldsymbol{x})$) approximating the posterior's mean and variance, both learned using the reparametrization trick. In the figure, we can also observe how our model's encoder comprises the Transformer backbone and VI network.

Decoder As for the encoder, the first component of the decoder is the "standard" Transformer decoder of Vaswani et al. (2017) and uses the same parameters as the Transformer encoder.

The Transformer decoder's last layer output is passed to a 128-component MDN, with trainable parameters ϕ and π: ϕ encodes the mean and variance of each multivariate gaussian components; π contains the probabilities of the categorical distribution that generates the mixing coefficient for each component. Concisely, we estimate a posterior as a series of M posteriors parameterized by $\langle \phi, \pi \rangle$, i.e. $\tilde{q}_m^{\phi;\pi}(z_{dec}|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$, conditioned via the decoder's Transformer, on both the gated latent encoder's output and previous pre-

[3] https://github.com/josecar25/MDN-VNMT

[4] We used the implementation of https://github.com/riannevdberg/sylvester-flows

Corpus	#sentences	#tokens	ASL	TTR	#chars
train set					
WMT	2.2M	64.2M	29.7	0.20	335
OpenSub.	9.2M	57.7M	6.73	0.18	428
test set					
OpenSub.	11,000	66,148	6.01	0.23	111
newstest	3,003	68,155	22.70	0.23	111

Corpus	#sents	#tokens	ASL	TTR	#chars
UGC test					
PFSMB	777	13,680	17.60	0.32	116
MTNT	1,022	20,169	19.70	0.34	122
UGC blind					
PFSMB	777	12,808	16.48	0.37	119
MTNT	599	8,176	13.62	0.38	127
4Square	1,838	18,234	9.92	0.22	109

Table 1: Statistics on the French side of the corpora used in our experiments. TTR stands for Type-to-Token Ratio, ASL for average sentence length and #chars for the number of different characters.

dicted tokens, $\boldsymbol{y}_{1:t-1}$. The MDN's mixing coefficient ($\alpha_m(\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$) network also takes the same input and is computed separately, by either using a fully-forward layer with $\mathrm{softmax}$ activation or the relaxed categorical Gumbel distribution. Both networks computing $\tilde{q}_m$ and α_m are jointly trained in an end-to-end fashion, such that translation loss is minimal for representations sampled from the resulting mixture, obtained according to Equation 4.

4 Training models

All systems are trained using a batch size of 4,096 tokens using the Adam optimizer (Kingma and Ba, 2015) accumulating gradients every 2 steps, and the `Noam` learning rate schedule (Vaswani et al., 2017) with 8K warmup steps. Throughout training, learning rate attains a maximum of $7\mathrm{e}{-4}$ and minimum of $1\mathrm{e}{-5}$. Both encoder and decoder Transformers are trained using 0.1 dropout and we employed 0.1 label smoothing (Szegedy et al., 2016). Training for, at most, 300K training iterations on a single Nvidia V100 took about 40 hours to converge for the `multi-VNMT` models, 34 hours for `VNMT-baseline` and 28 hours for the non-latent `Transformer` baseline. In order to avoid posterior collapse, and as done in Setiawan et al. (2020), we use β_C-VAE (Prokhorov et al., 2019), with values $\beta = 1$ and $C = 0.1$. Additionally, we used a Kullback-Leibler (KL) annealing schedule of 100K iterations for training. We set a 10% probability of dropping the target word (Bowman et al., 2016). We have chosen, as initial experimental configuration, $\tau = 1.0$ for the Gumbel-Softmax sampling temperature, which was selected mainly aiming to avoid artificial gradient scaling during backpropagation (c.f. Equation (5)). A beam of width 5 has been used for evaluation.

5 Experiments

Datasets We train our different MT models on two different French to English canonical parallel corpora: the first one is a subset of the `WMT` corpus, i.e. Europarl (v7) and NewsCommentary(v10) (Bojar et al., 2015) and the second one is the`OpenSubtitles`'18 corpus (Lison et al., 2018). We used BPE tokenization (Sennrich et al., 2016) with 16K merge operations.

Detailed statistics on our corpora can be found in Table 1.

UGC Test Sets To evaluate the different NMT models, we consider two data sets of manually translated UGC: `MTNT` (Michel and Neubig, 2018) and the Parallel French Social Media Bank corpus (`PFSMB`) (Rosales Núñez et al., 2019)[5] which extends the French Social Media Bank (Seddah et al., 2012) with English translations. These two data sets raise many challenges for MT systems: they notably contain characters that have not been seen in the training data (e.g. emojis), rare character sequences (e.g. inconsistent casing or usernames) as well as many OOVs denoting URL, mentions, hashtags or more generally named entities (NE). Most of the time, sOOVs are exactly the same in the source and target sentences.

We also consider the `4Square` corpus (Berard et al., 2019) as a blind test to validate our conclusions. To analyze our neural representations (§7), we use a subset of the `PFSMB`, called `PMUMT`, which contains 400 annotated and normalized French to English UGC sentences (Rosales Núñez et al., 2021).

Protocols Translation quality was evaluated using BLEU (Papineni et al., 2002) and `chrF2` (Popovic, 2017) both computed by SACREBLEU

[5] `https://gitlab.inria.fr/seddah/parallel-french-social-mediabank`

(Post, 2018) with the 'intl' tokenization, after detokenizing the systems outputs.

In all the experiments we used the hyper-parameters values reported by Vaswani et al. (2017) and only choose the number of components of the MDN and the dimension of the latent representation on the validation set.[6] Regarding the latent dimension, we conducted the same study with 128, 256 and 512 dimensions, with 128 being the best value. A beam of size 5 has been used for evaluation.

6 Results

In this section we present the main MT results to study MT performance of our methods.

MT Performance Our first experiment aims to compare the performance of multi-VNMT, the model we introduced in Section 3, to that of a "vanilla" Transformer model and of a state-of-the-art VNMT system using NF.[7] The first baseline, a non-latent NMT architecture, Transformer, corresponds to our model without its VI components (i.e. with only the Transformer encoder and decoder); the second baseline, VNMT-baseline, corresponds to the equivalent of our NF setup (featuring 4 Planar Flows) from Setiawan et al. (2020).

Results achieved by these systems are reported in Table 2. We computed the 95% statistical significance by using a 1,000-samples bootstrapping, as in Koehn (2004). It should first be noted that the performances of the three systems we consider are identical when they are evaluated on in-domain data, whatever the evaluation measure considered (no statistically significant difference between the models). This observation highlights one of the strength of the proposed method: contrary to fine-tuning (arguably the most common method to adapt a system to a new domain) that often hurts performance on in-domain evaluation because of catastrophic forgetting (McCloskey and Cohen, 1989), the improvement of the quality of UGCs by the proposed method is not at the expense of the quality of translation of canonical texts.

It also appears that, on out-of-domain text, multi-VNMT, the approach proposed in this work, outperforms the standard Transformer model as well as the state-of-the-art VNMT model, supporting our hypothesis that considering several variational inference components allows to better capture all the variations that can be found in UGC and will result in improved translation quality. Interestingly, our system also performs better than Transformer when evaluated on out-domain canonical data and not only on UGC data. It should be noted, however, that the gains of our model are consistent but small and statistically significant mainly when translation quality is evaluated with chrF2.

Ablation study To better understand the impact of the different components of our model, we conduct an ablation study whose results are reported in Table 3. Overall, we obtain the best BLEU scores across all test sets for the "full" multi-VNMT model.

In particular, it appears that *static* latent representation (z static in Table 3), where instead of sampling from the learned distributions, we retrieve their mean as output, show slightly better BLEU scores when translating the MTNT with the model trained on OpenSubtitles and the newstest'14 test set with the model trained on WMT (+0.1 improvement in the two cases). However, results are inconsistent for UGC test sets and otherwise worse than those of the full model for both in-domain and canonical OOD test sets for our two training configurations. This might be explained by the lack of stochastic perturbations provided by the sampling step during training, leading the model to lose generalization during evaluation.

It is also interesting to note that using a categorical variational version of the mixing coefficients rather than the usual choice of computing them with a softmax improves translations quality: the latter is only performing better for the newstest'14 test set when training on the OpenSubtitles corpus (π *non-latent*). Following the same trend, the WMT training data configuration also show improvements when using the Gumbel-Softmax version, for which +0.8 and +0.3 BLEU point improvement were obtained for both the PFSMB and MTNT UGC testsets, respectively.

Posterior collapse We have computed the average KL divergence of the variational decoder's block (i.e. $D_{KL}\left(q_\phi(z|x,y)||p_\theta(z|x)\right)$ on the encoder side) of multi-VNMT and its ablated ver-

[6] For the number of components we tested the following values 8, 16, 32, 64, 128 and 256 and found the optimal value to be 128.

[7] We re-implemented the system of Setiawan et al. (2020).

		WMT				OpenSubtitles				
		PFSMB †	MTNT †	News◇	OpenSubTest	PFSMB †	MTNT †	News	OpenSubTest ◇	# params.
BLEU	Transformer	15.1	21.3	**27.9**	16.4	27.7	28.4	26.4	31.4	69M
	VNMT-baseline	15.5	21.4	**27.9**	16.4	28.0	28.9	**26.5**	31.4	72M
	multi-VNMT	**16.0***	**21.8**	**27.9**	**16.7***	**28.4**	**29.2**	26.4	**31.5**	77M
chrF2	Transformer	37.8	45.1	54.4	38.6	46.9	48.3	52.6	48.9	69M
	VNMT-baseline	38.3	45.1	**54.6**	38.6	47.6	49.2*	**53.1***	48.9	72M
	multi-VNMT	**38.5***	**45.5**	**54.6**	**39.0***	**47.7***	**49.6***	52.9*	**49.0**	77M

Table 2: BLEU and chrF2 test scores for our models. The † symbol indicates the UGC test sets, and ◇ in-domain test sets. Highest metric for each test set are in bold; scores significantly better than Transformer ($p < 0.05$) are marked with a *.

	WMT				OpenSubtitles				
	PFSMB †	MTNT †	News◇	OpenSubTest	PFSMB †	MTNT †	News	OpenSubTest ◇	# params.
multi-VNMT	16.0	**21.8**	27.9	**16.7**	**28.4**	29.2	26.4	**31.5**	77M
π non-latent	15.8	21.0	27.8	16.4	28.1	28.5	**26.6**	31.3	77M
-NF	15.3	21.6	**28.0**	16.5	28.3	28.8	26.1	31.3	76M
Z STATIC	**16.5**	20.9	**28.0**	16.4	28.1	**29.3**	26.2	31.4	76M
-MDN	**16.5**	20.9	27.8	16.6	27.7	28.7	26.2	31.3	73M

Table 3: BLEU test scores our ablated variants. The † symbol indicates the UGC test sets, and ◇ in-domain test sets.

sion without the MDN module in an in-domain setting. When trained (using the same KL annealing schedule) on OpenSubtitles (resp. WMT) this divergence is 0.21 (resp. 0.38) for multi-VNMT and 0.15 (resp. 0.33) when removing the MDN block, suggesting that our proposed architecture is less prone to suffer from the posterior collapse phenomenon.

7 Analyzing Latent Representations

In this Section, we describe several experiments aiming at understanding how multi-VNMT uncovers more robust representations than the VNMT baseline.

Impact of Noise in the Source First, to evaluate the perturbations that the model suffers when noise is present in the source, we measure the cosine similarity between the representations of the French noisy sentences and their normalized version, taking advantage of the PMUMT corpus (§5). More precisely, we compare the source-side embeddings of the 400 original noisy UGC sentences and their corresponding 400 fully-normalized versions built by VNMT-baseline and multi-VNMT. We observe that the average cosine similarity between the noisy and normalized learning representations of multi-VNMT is 0.36 compared to an average similarity of 0.26 for the representations of VNMT-baseline, suggesting that the former provides more robust representations of UGC than the latter, a conclusion supported by the distribution of similarities shown in Figure 2.

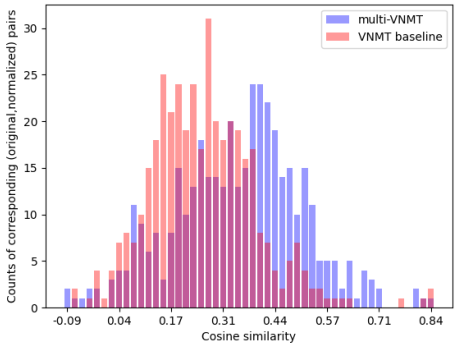
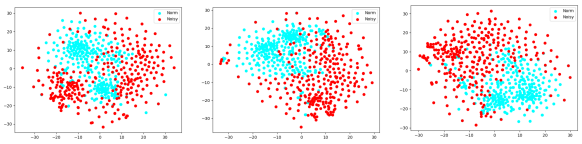


Figure 2: Distribution of cosine similarities between the representations of noisy and normalized sentences of PMUMT built by the encoder of VNMT-baseline and multi-VNMT.

Noisy vs normalized data To complete the previous analysis, we have reported, in Figure 3, the projection of the representations of noisy and normalized sentences computed by t-SNE. We can notice how both VNMT systems have a tendency to separate noisy and normalized sentences compared to Transformer, while both having higher cosine similarity than the latter.



(a) Transformer (b) VNMT-baseline (c) multi-VNMT

Figure 3: t-SNE projection of the encoder source embeddings for noisy sentences and their normalized versions.

	PFSMB †	MTNT †	News	OpenSub.◇
Transformer	27.7	28.4	26.4	**31.4**
Pre-trained init.	**29.0**	28.2	26.2	31.3
Frozen embs.	28.4	**28.9**	**26.8**	31.3
Fine-tuned	28.4	**28.9**	26.5	**31.4**

Table 4: Using VNMT-learned embeddings for transfer robust learned representations to the Transformer. The † symbol indicates the UGC test sets, and ◇ in-domain test sets.

Transferring learning representations As discussed above, in Figure 3 we noticed that VNMT seems to enforce noisy morphology modeling to the Transformer's embeddings in an implicit way. This motivated us to study whether the information in such learning representations can be used by the Transformer backbone model and benefit from improved robustness while removing the direct latent space contribution, and notably, with the same number of parameters and architecture as Transformer. Thus, in Table 4, we report BLEU scores for the Transformer model trained on OpenSubtitles, by either initializing the VNMT-pretrained source-side embeddings before training, or fine-tuning (FT) the system. We have performed FT using the same data configuration as in OpenSubtitles and continued training for 3 epochs from the Transformer model in Table 2 while replacing the Transformer's source embeddings by their VNMT-learned version's weights.

Results in Table 4 provide evidence that VNMT enforces more robust embeddings, which perform consistently better over the PFSMB UGC test set compared to the baseline, the system Frozen embs giving the most consistent results over UGC. This system also achieves the best newstest'14 canonical OOD test set in the OpenSubtitles setup, while taking advantage of an increased robustness to UGC. These results

	PFSMB (Blind)	MTNT (Blind)	4Square
Transformer	19.7	25.0	21.9
+FT emb.	19.4	25.3	22.0
VNMT-baseline	**20.0**	25.3	22.0
multi-VNMT	**20.0**	**26.4**	**22.5**

Table 5: BLEU scores of our best systems on blind test sets.

indicate that our VNMT model leads to embeddings that are more robust to noise even when used in a classic transformer-based NMT baseline. An interesting path of research would be to evaluate these embeddings in other tasks and scenarios (e.g. Cross lingual UGC Q&A).

8 Blind test sets scores

We evaluated our best performing model (multi-VNMT trained on OpenSubtitles) on the blind test sets described in § 5, translating another set of tests to assess whether our approach proves useful for generalization over different types of UGC. We have also included the 4Square corpus (Berard et al., 2019) to validate our VNMT system on other domain of UGC (restaurant reviews). We also display the results when using the VNMT-baseline baseline and the Transformer model to assess improvement of our proposed architecture. We report such results in Table 5, where we can see that, when translating our blind UGC test sets, multi-VNMT consistently outperforms the baselines. It is interesting to notice that, although the in-domain performances for these 3 systems are very similar (between 31.4 and 31.5 BLEU in Table 2), the performance gap of blind UGC test sets is larger, i.e. +0.8 BLEU in average compared to the non-latent baseline.

9 Discussion

How MDN behaves under noise In Appendix A, we discuss how MDN components are activated when translating canonical in-domain and OOD texts, as well as UGC and normalized UGC. In Figure 4 and Table 6 in the Appendix, we show that noisy UGC activates MDN's components with low correlation to other OOD canonical texts and even to its normalized version, which implies that the distribution of the kernels' mixing coefficients is relatively among, the 4 test sets

considered, unique, i.e. relatively uncorrelated from the activation of other canonical texts (in-dom and OOD), when processing UGC. We cannot conclude, however, whether this observation is a consequence of the noise propagated through the model's networks, but the enhanced robustness we witnessed in the translation results (much better performance to UGC, while keeping on-par or slightly better canonical (in-domain and OOD) performance) suggests that these mixing coefficients (that ultimately control the final decoding output) activate different variational posteriors (one per kernel) that can better process UGC.

Conclusions We introduced a novel VNMT architecture that provides improved performance and robustness over a state-of-the-art VNMT model, specifically when translating French UGC. An ablation study and blind test sets evaluation validate our architecture choice in regards of robustness capabilities for such texts. In addition, by exploring the learning representations trained by our VNMT model, and through conducting transfer learning experiments with such, we investigate the robustness brought to UGC, and show that VNMT enforces such property to the backbone model, bringing a promising avenue for more robust pre-trained neural learning representations. However, an open question arising from this work, it is currently unclear if the performance gain we observed is due to a better generalisation to distributional shift or if it corresponds to a better adaptation to noise in the input. Future works will be devoted to this question, which can be abstracted away to study whether UGC idiosyncrasies are a form of noise, some parts being learnable, or are rather points to a new domain.

Acknowledgments

We thank our anonymous reviewers for providing insightful comments and suggestions. This work was funded by the ANR projects ParSiTi (ANR-16-CE33-0021). This work was granted access to the HPC resources of IDRIS under the allocation 2021-AD011011748R1 made by GENCI.

References

Alexandre Berard, Ioan Calapodescu, Marc Dymetman, Claude Roux, Jean-Luc Meunier, and Vassilina Nikoulina. 2019. Machine translation of restaurant reviews: New corpus for domain adaptation and robustness. In *Proceedings of the 3rd Workshop on Neural Generation and Translation@EMNLP-IJCNLP 2019, Hong Kong, November 4, 2019*, pages 168–176. Association for Computational Linguistics.

Rianne van den Berg, Leonard Hasenclever, Jakub M. Tomczak, and Max Welling. 2018. Sylvester normalizing flows for variational inference. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 393–402. AUAI Press.

Christopher M. Bishop. 1994. Mixture density networks. Technical report.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT@EMNLP 2015, 17-18 September 2015, Lisbon, Portugal*, pages 1–46. The Association for Computer Linguistics.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 10–21. ACL.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 359–369.

Jennifer Foster. 2010. "cba to check the spelling": Investigating parser performance on discussion forum posts. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 381–384, Los Angeles, California. Association for Computational Linguistics.

Emil Julius Gumbel. 1954. *Statistical theory of extreme values and some practical applications; a series of lectures*. Applied mathematics series ; 33. U.S. Govt. Print. Office, Washington.

David Ha and Douglas Eck. 2018. A neural representation of sketch drawings. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

David Ha and Jürgen Schmidhuber. 2018. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*

31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pages 2455–2467.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Yoon Kim, Sam Wiseman, and Alexander M. Rush. 2018. A tutorial on deep latent variable models of natural language. *CoRR*, abs/1812.06834.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander M. Rush. 2018. Opennmt: Neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas, AMTA 2018, Boston, MA, USA, March 17-21, 2018 - Volume 1: Research Papers*, pages 177–184.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. pages 388–395.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. Opensubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Héctor Martínez Alonso, Djamé Seddah, and Benoît Sagot. 2016. From noisy questions to Minecraft texts: Annotation challenges in extreme syntax scenario. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 13–23, Osaka, Japan. The COLING 2016 Organizing Committee.

Arya D. McCarthy, Xian Li, Jiatao Gu, and Ning Dong. 2020. Addressing posterior collapse with mutual information for improved variational neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8512–8525. Association for Computational Linguistics.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press.

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 543–553.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318.

Maja Popovic. 2017. chrf++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 612–618. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191.

Victor Prokhorov, Ehsan Shareghi, Yingzhen Li, Mohammad Taher Pilehvar, and Nigel Collier. 2019. On the importance of the Kullback-Leibler divergence term in variational autoencoders for text generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 118–127, Hong Kong. Association for Computational Linguistics.

Michael Przystupa. 2020. *Investigating the impact of normalizing flows on latent variable machine translation*. Ph.D. thesis, University of British Columbia.

Danilo Jimenez Rezende and Shakir Mohamed. 2015. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1530–1538. JMLR.org.

José Carlos Rosales Núñez, Djamé Seddah, and Guillaume Wisniewski. 2019. Comparison between NMT and PBSMT performance for translating noisy user-generated content. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 2–14, Turku, Finland. Linköping University Electronic Press.

José Carlos Rosales Núñez, Djamé Seddah, and Guillaume Wisniewski. 2021. Understanding the impact of UGC specificities on translation quality. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 189–198, Online. Association for Computational Linguistics.

Sergej Schmunk, Wolfram Höpken, Matthias Fuchs, and Maria Lexhagen. 2013. Sentiment analysis: Extracting decision-relevant knowledge from ugc. In *Information and Communication Technologies in Tourism 2014*, pages 253–265, Cham. Springer International Publishing.

Djamé Seddah, Benoît Sagot, Marie Candito, Virginie Mouilleron, and Vanessa Combet. 2012. The french social media bank: a treebank of noisy user generated content. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pages 2441–2458.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Hendra Setiawan, Matthias Sperber, Udhyakumar Nallasamy, and Matthias Paulik. 2020. Variational neural machine translation with normalizing flows. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7771–7777. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

E. G. Tabak and Cristina V. Turner. 2013. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Tim Z. Xiao, Aidan N. Gomez, and Yarin Gal. 2020. Wat zei je? detecting out-of-distribution translations with variational transformers. *CoRR*, abs/2006.08344.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. 2016. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 521–530. The Association for Computational Linguistics.

Hao Zheng, Zhanlei Yang, Wenju Liu, Jizhong Liang, and Yanpeng Li. 2015. Improving deep neural networks using softplus units. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–4.

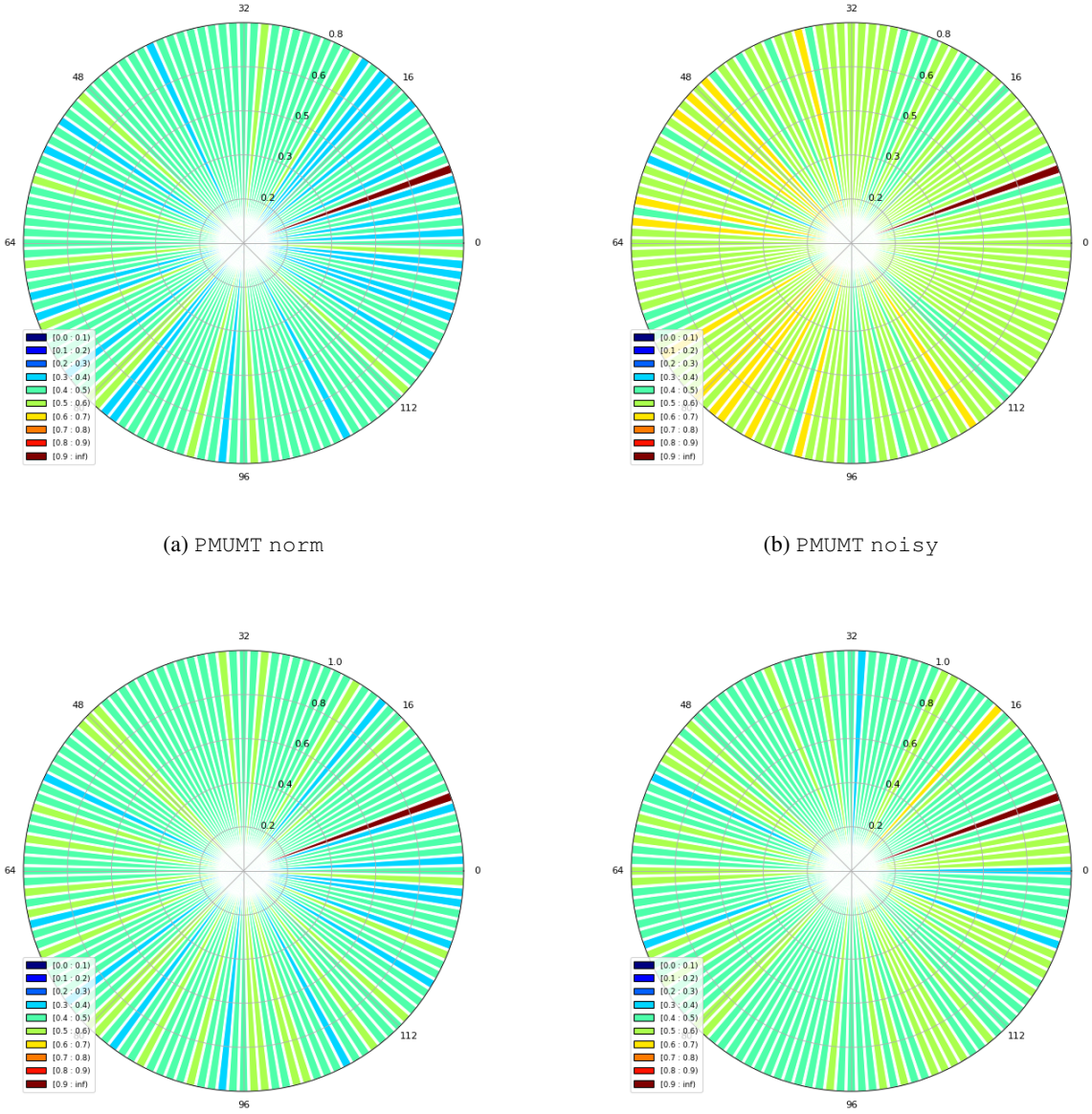
A How do MDN's components react to UGC?

We proceeded to analyze and visualize how the MDN mixture coefficients react when translating our different test sets. In order to do so, in Figure 4 we report results for the canonical test sets, the normalized `PMUMT` corpus, and its noisy original UGC version. Each bar of the Wind Rose diagram represents one of the 128 independent trained distributions' mixture weights, which have been normalized and scaled across the four graphics, and where the 7th MDN component seems to be consistently the one that drives most of the decoding for the presented experiments. Furthermore, we can notice that most mixing coefficients are, for the most part, have around 50% probability of contributing to the final inference mixture, despite not enforcing this behavior with any specific method (e.g. dropout). On the other hand, the visualization suggests that both yellow (50-60%) and blue components (30-40% of activation) are variable across test sets, being very similar between `PMUMT Norm` and `OpenSubTest`, which could indicate that the mixture components are learning to encode different types of texts, potentially working as an implicit topic modeling module. Regarding the visualization when translating `PMUMT Noisy`, the main MDN component identified above, seems less important even when compared to the out-of-domain `newstest'14` set, which suggests that the MDN uses more dense representations when processing noisy texts.

In parallel, in Table 6 we display the covariance of these coefficients' distributions between the combinations of their values when translating different kind of texts, along with the standard deviation and sparsity to describe how the MDN's components behave.

Comparing the visualization in Figure 6, we can notice how the noisy UGC `PMUMT` and the out-of-domain `newstest'14`, diverge from the in-domain `OpenSubTest` and normalized UGC `PMUMT` corpus. This correlation is evidenced in the results in Table 6, where `PMUMT` noisy has the lowest score when compared to every other corpus, even if its normalized version seems to

be the most correlated to the in-domain evaluation. Specifically, PMUMT Noisy is the least correlated to in-domain OpenSubTest and out-of-domain newstest'14 corpora, which points to the MDN reacting differently to content domain and UGC specificities in the noise; this observation is also supported by the associated figure. It is also interesting to notice that, according to the standard deviation and sparsity values, the active MDN components are more dense and variable for out-of-domain evaluation conditions, for the same Gumbel sampling temperature value.



(a) PMUMT norm

(b) PMUMT noisy

(c) OpenSubTest

(d) newstest'14

Figure 4: Average MDN mixture weights for test sets of different natures.

	PMUMT Noisy	News	OpenSubTest	std.	sparsity
PMUMT Norm	8.16	9.71	13.05	1.2e-3	0.387
PMUMT Noisy	—	7.72	7.86	1.0e-3	0.382
News	—	—	9.42	1.1e-3	0.384
OpenSubTest	—	—	—	1.1e-3	0.387

Table 6: Covariance between MDN mixture coefficients during inference for different types of test sets and sparsity for each set. *std.* stands for the standard deviation.

Multilingual Automatic Speech Recognition for Scandinavian Languages

Rafal Cerniavski
Uppsala University
Conversy AB
rafal.cerniawski@gmail.com

Sara Stymne
Department of Linguistics and Philology
Uppsala University
sara.stymne@lingfil.uu.se

Abstract

We investigate the effectiveness of multilingual automatic speech recognition models for Scandinavian languages by further fine-tuning a Swedish model on Swedish, Danish, and Norwegian. We first explore zero-shot models, which perform poorly across the three languages. However, we show that a multilingual model based on a strong Swedish model, further fine-tuned on all three languages, performs well for Norwegian and Danish, with a relatively low decrease in the performance for Swedish. With a language classification module, we improve the performance of the multilingual model even further.

1 Introduction

Automatic speech recognition (ASR) is the task of transforming speech into text, often referred to as transcription. Multilingual ASR tackles the task in multiple languages with the same model or pipeline. Modern ASR architectures such as DeepSpeech (Hannun et al., 2014), Wav2Vec (Baevski et al., 2020), and Whisper (Radford et al., 2022) are capable of transcribing speech with Word Error Rates (WERs) well below 10 percent. To achieve this, models require copious amounts of data, which is unavailable for the vast majority of languages. For low-resource languages, multilingual models as means of bootstrapping the performance are often the only solution.

Conneau et al. (2021) demonstrate that a multilingual setting can be beneficial even for high-resource languages. Pratap et al. (2020), however, suggest that limiting models to smaller, typologically related languages is more productive than training on all languages at once. As such, it can be argued that Scandinavian languages are a great fit for multilingual NLP models. Swedish, Danish, and Norwegian all originate from old Norse and share numerous similarities, such as largely overlapping lexicons and similar grammar. As noted by Delsing and Lundin Åkesson (2005), the similarities across the three languages are not linear, since Swedish and Norwegian are most similar in speech, whereas Danish and Norwegian are most similar in writing. Nevertheless, Sahlgren et al. (2021) argue that Scandinavian languages are so similar that large text-based language models for these languages should be trained jointly. It has also been shown that utilizing the similarities between the Scandinavian languages can improve text-based tasks such as machine translation (e.g. Tiedemann, 2009) and parsing (e.g. Smith et al., 2018). However, to the best of our knowledge, there is no work where the usefulness of combining the Scandinavian languages is reported for speech-based tasks, such as ASR.

We focus on identifying whether a multilingual ASR model for Swedish, Danish, and Norwegian can be trained to utilize an existing high-quality monolingual model, as we fine-tune a strong Swedish end-to-end model to also handle the Danish and Norwegian languages. In addition, we analyze how well the monolingual ASR models transfer across the Scandinavian languages in a zero-shot setting. We also evaluate how the multilingual setting affects the quality of transcription as opposed to monolingual settings. Lastly, we show that a language classification module can be used for selecting a language model in the multilingual setting. We conduct all experiments on the Wav2Vec 2.0 (Baevski et al., 2020) based ASR models. Additional experiments, as well as more in-depth analysis, can be found in Černiavski (2022).

2 Previous Work

Language Models for ASR The usage of a language model in speech recognition has contin-

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 460–466

uously proven to boost the quality of transcription. Positive results have been observed with both statistical n-gram language models (Amodei et al., 2016; Håkansson and Hoogendijk, 2020) and transformer-based models, such as BERT (Baevski et al., 2020). Most considerable improvements seem to result from domain-specific language models, as contextualization and biasing of models have repeatedly improved the quality of transcription (Aleksic et al., 2015).

Multilingual ASR Transcription of multiple languages via a single model or pipeline has been made possible through a variety of architectures. Approaches range from an assemble of monolingual models connected through a preceding language classification component (Lyu and Lyu, 2008; Mabokela and Manamela, 2013; Barroso et al., 2010), to models sharing the phone models (Lin et al., 2009) or hidden layers of acoustic models (Yu and Deng, 2015) across two or more languages, to being conjunct on all levels, sharing all components and treating all input languages as one (Pratap et al., 2020; Conneau et al., 2021).

As a general rule, the effects of a multilingual setting on the quality of transcription are twofold. Low-resource languages tend to reap the benefits, as models seemingly generalize from the patterns learned in higher-resource languages (Yu and Deng, 2015; Bhable and Kayte, 2020). High-resource languages, however, tend to suffer (Lin et al., 2009; Conneau et al., 2021), likely due to the noise introduced through the exposure of models to data in (a) foreign language(s). Nevertheless, Pratap et al. (2020) demonstrated that there appear to be ways of mitigating the toll of a multilingual setting on the resource-rich languages by means of a typologically motivated choice of languages in a cluster as well as cluster-specific rather than one-for-all decoders.

3 Methodology

We first evaluate the performance of monolingual Swedish, Danish, and Norwegian models on the test sets of each language (i.e. the Swedish model was evaluated on Swedish, Danish, and Norwegian test sets). For comparison, we also evaluate the performance of an English ASR model on the three Scandinavian languages. We do so first to obtain comparable word error rates of each model for their intended language, except for English; second, to explore a zero-shot setting, where we explore whether the typological similarity of Scandinavian languages enables the ASR models trained on one of the languages to transcribe the data in the other two languages. We add English, a more distant Germanic language, for comparison.

In a second set of experiments, we fine-tune trial multilingual ASR models for Swedish, Danish, and Norwegian. We aim to utilize the high quality of the already fine-tuned Swedish model (Malmsten et al., 2022) to bootstrap the transcription in Danish and Norwegian as opposed to training a model on the three languages from scratch. As such, we attempt fine-tuning the Swedish model in the following settings:

1. **Retraining DA+NO** - using complete training sets in Danish and Norwegian, with no Swedish training data (30,000 entries total)
2. **Retraining DA+NO+SE_half** - using complete training sets in Danish and Norwegian, and half of the Swedish training data (37,500 entries in total)
3. **Retraining DA+NO+SE_full** - using complete training sets of all three languages (45,000 entries in total)

For comparison, we also train a model on all three languages (15,000 entries per language, 45,000 total) on the pre-trained, but not fine-tuned Swedish model[1]. We train these models for 5 epochs and evaluate on the trilingual development set every 1,000 updates. In order to investigate the effect of adding a language model for the multilingual models, we train a language classifier, see Section 5 for details.

In our final experiment, we select the best-performing trial model (**Retraining DA+NO+SE_full**) to train a multilingual model for 20 epochs. We evaluate the model in two settings. In the first, we use no language model in the decoding. In the second, we use the language classifier to predict the language, in order to select a 4-gram language model for the predicted language. We train the 4-gram language models on the entirety of the original NST training sets, except for Swedish, where we exclude the sample used as test set.

We report Word Error Rate as our main evaluation metric and perform a brief qualitative analysis of the most common errors.

[1] https://huggingface.co/KBLab/wav2vec2-large-voxrex

4 Data and Models

Data We created testing, training, and development subsets for Swedish, Danish, and Norwegian from two datasets: Nordisk Språkteknologi (NST)[2] and Common Voice (CV) 8.0 (Ardila et al., 2020). For testing subsets of Danish and Norwegian, we used the entire test sets from NST, which amount to 77.1 and 115.3 hours of speech respectively. For Swedish, due to the lack of a modernized version of the NST test at the time of working on the project, we randomly sampled 20% of the training set - roughly 73,2 hours of speech. For training, we limit the subsets to 15,000 entries per language (roughly 7 hours of speech) per language due to limited computational resources. We ensure that the Swedish train and test subsets do not overlap.

We use the CV dataset to construct a validation set for the multilingual models. For Swedish and Danish, we randomly sampled 2,000 validated entries per language. No validated data were available for Norwegian Bokmål; we, therefore, used a held-out sample of 2,000 entries from the Norwegian NST training dataset. In total, we used roughly one hour of speech per language for validation.

We processed each subset by downsampling the audio to 16 kHz and normalizing the transcriptions. The normalization involved lower-casing all characters and removing non-alphanumeric characters, such as punctuation markers.

Models For monolingual baselines in Swedish, Norwegian, and English, we used Wav2Vec 2.0 models publicly available on Huggingface[3]. Due to the lack of an existing fine-tuned Danish model at the time, we fine-tuned one ourselves: we used the publicly available pre-trained Danish Wav2Vec 2.0 model[4], which we then fine-tuned on one GPU for 10 epochs on our Danish NST train subset. In the encoder, we retain the original parameters of the pre-trained model, whereas in the decoder, we set the batch size to 10, gradient accumulation steps to 3, learning rate to 1e-4, and weight decay to 0.005.

[2]Swedish: https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-56/;
Danish: https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-55/;
Norwegian: https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-54/

[3]Swedish: https://huggingface.co/KBLab/wav2vec2-large-voxrex-swedish
Norwegian: https://huggingface.co/NbAiLab/nb-wav2vec2-1b-bokmaal
English: https://huggingface.co/facebook/wav2vec2-base-960h

[4]https://huggingface.co/Alvenir/wav2vec2-base-da

Model	Test Set	No LM	With LM
Swedish	Swedish	2.19%	2.74%
	Danish	78.58%	72.69%
	Norwegian	61.78%	52.06%
Danish	Swedish	120.10%	98.93%
	Danish	19.14%	13.82%
	Norwegian	104.56%	90.06%
Norwegian	Swedish	83.51%	73.82%
	Danish	83.79%	75.05%
	Norwegian	16.47%	12.03%
English	Swedish	110.06%	93.59%
	Danish	99.50%	88.71%
	Norwegian	102.52%	90.35%

Table 1: WERs of monolingual models on the three Scandinavian languages, no language model versus a 4-gram language model.

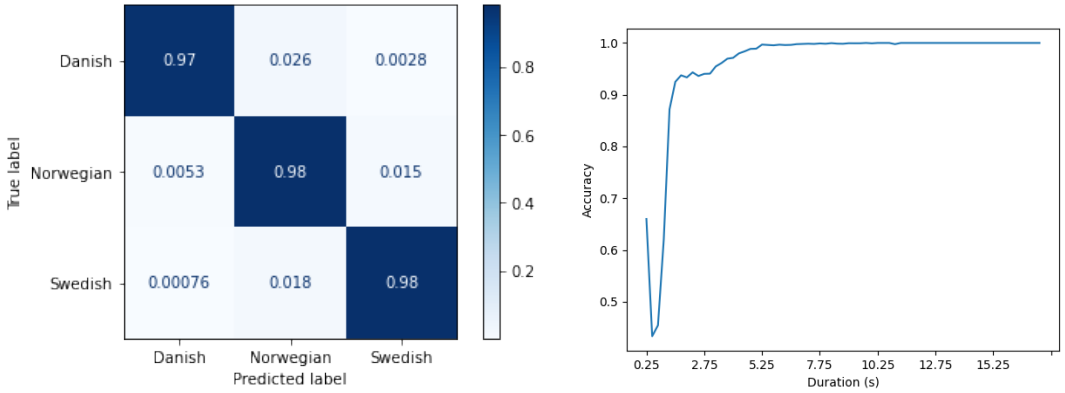
5 Language Classification Module

The language classification module is initialized on top of the same pre-trained Swedish Wav2Vec 2.0 model we used for ASR. We train it on 15,000 entries per language, randomly sampled from the train sets (45,000 entries in total). We set the batch size to 4, learning rate to 1e-4, and gradient accumulation steps to 2, and use mean pooling.

We evaluate the classification module on a concatenation of the test sets from all three languages, with results in Figure 1. The classifier reached an overall accuracy of 98% across the three languages, with very few confusions between Danish and Swedish. It is also noticeable that most errors occur for short segments, often containing a single word. For segments of at least five seconds, the accuracy is near perfect.

6 Results and Discussion

WERs for Swedish, Danish, Norwegian, and English ASR models on the three Scandinavian languages are shown in Table 1. The results on zero-shot ASR are poor. We can see some general patterns in the performance across languages. The Swedish and Norwegian models perform better for all three Scandinavian languages than the English model. However, the Danish model performs as poorly on Swedish and Norwegian data as the En-



(a) Classification Matrix

(b) Accuracy and Duration

Figure 1: Evaluation of the language classification module. The accuracy in subgraph b is averaged over the three languages.

glish model. This could stem from the fact that our Danish model was trained on very little data compared to the Swedish and Norwegian models, as the Danish model performs poorly even on the Danish data despite being trained on in-domain data. However, it could be affected by the fact that the pronunciation in Danish is quite different from Swedish and Norwegian.

Even though the results are poor, we note that the results for the Scandinavian languages largely follow the patterns for mutual intelligibility between human speakers (Delsing and Lundin Åkesson, 2005); the Swedish ASR model is better at transcribing Norwegian than Danish, the Danish model is better for Norwegian than Swedish, and the Norwegian model is somewhat better for Swedish than Danish. The latter difference is more pronounced for character error rates, see (Černiavski, 2022).

The scores confirm that the addition of a simple n-gram language model leads to stable improvements of the quality in transcription, even in a cross-lingual setting. The Swedish model is an exception, though, likely due to the overall high quality of the model, which is only limited by such a language model.

Lastly, qualitative analysis of the outputs reveals that some of the predictions considered to be errors due to a deviation from the ground truth are grammatically correct alternative spellings that can have the same pronunciation. For instance, in the output of the monolingual Swedish model, some of the most common substitution errors are *skall* instead of *ska*, *i stället* instead of *istället*, and *i dag* instead of *idag*. Due to the usage of WER as an evaluation metric, the latter two examples are treated as 2 errors each. This is because WER considers *istället* to be substituted with *stället* and treats the preposition *i* to be an insertion error. Similar patterns can be observed in the outputs for the other two languages, which leads us to believe that WER might not be the most suitable evaluation metric for Scandinavian, and possibly other, languages.

WER for the trial multilingual models are shown in Figure 2. The results indicate that the initialization of the multilingual model from a monolingual model is only effective in low-resource settings. This is because a model trained from scratch on all three languages reaches comparable WER within roughly 5,000 steps. Nevertheless, despite the subtle difference, the average WERs on all three languages indicate that the model initialized from a fine-tuned Swedish model and further fine-tuned on complete training sets (**Retraining DA+NO+SE_full**) is second only to monolingual baselines. Analogous patterns can be seen in terms of character error rates (Černiavski, 2022). Hence, we choose this setting for training our final multilingual model.

The scores of the final multilingual ASR model able to transcribe Swedish, Danish, and Norwegian, as opposed to the monolingual baselines are shown in Table 2. Using a language classification model for selecting which language model to add, leads to improvements for all three languages. We observe stable improvement over monolingual baselines for Norwegian and Danish both with

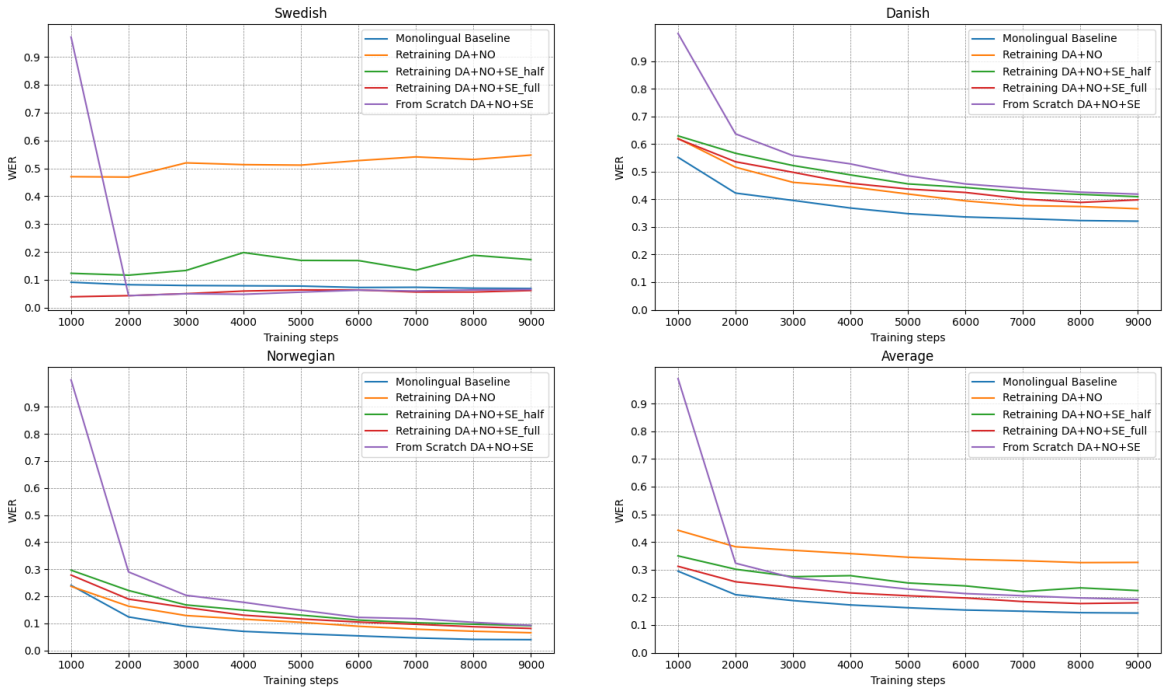


Figure 2: Word Error Rates of the multilingual trial models and a monolingual baseline on the evaluation set, mapped over training steps.

and without language models, with only a slight drop in performance for Swedish. However, more analysis is needed to investigate the influence of matching the language versus matching the domain since both our training and test sets are from the NST dataset.

We observe that the multilingual model performs significantly better in Norwegian than it does in Danish, which can also be seen from the progression in the WERs of the trial models shown in Figure 2. This is likely because the development and test sets for Norwegian we used were from the same domain, which was not the case for Danish, but it may also be influenced by Norweigan pronunciation being closer to Swedish than Danish to Swedish. Černiavski (2022) presents a more detailed qualitative analysis of the transcription in the monolingual versus multilingual setting, as well as with and without LM settings. We observe that cross-lingual errors (e.g. when a Swedish word is transcribed with a Norwegian spelling) are very rare in a multilingual setting even when LMs are not used.

Test Set	Model	No LM	With LM
Swedish	Mono	**2.19%**	2.74%
	Multi	4.61%	3.26%
Danish	Mono	19.14%	13.82%
	Multi	12.69%	**10.43%**
Norwegian	Mono	16.47%	12.03%
	Multi	9.64%	**6.51%**

Table 2: The performance **Mono**lingual baselines versus our **Multi**lingual model.

7 Conclusions

Multilingual automatic speech recognition is often considered to be useful only for low-resource languages. Though a multilingual model can hardly compete in ultra-high-resource languages, we show that the multilingual Scandinavian model can perform comparably or even perform better than monolingual models. Our results indicate that it could be useful to combine the Scandinavian languages not only for text, but also for speech processing. More extensive evaluation of models is needed to conclude whether the model benefits from a multilingual setting, or only from in-domain training. Further research could also explore the effects of a multilingual setting on the ability to classify dialects of Scandinavian languages.

Acknowledgments

Computations were enabled by resources in project UPPMAX 2020/2-2 at the Uppsala Multidisciplinary Center for Advanced Computational Science.

References

Petar S. Aleksic, Mohammad Reza Ghodsi, Assaf Hurwitz Michaely, Cyril Allauzen, Keith B. Hall, Brian Roark, David Rybach, and Pedro J. Moreno. 2015. Bringing contextual information to Google speech recognition. In *Interspeech*.

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, Jie Chen, Jingdong Chen, Zhijie Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Ke Ding, Niandong Du, Erich Elsen, Jesse Engel, Weiwei Fang, Linxi Fan, Christopher Fougner, Liang Gao, Caixia Gong, Awni Hannun, Tony Han, Lappi Johannes, Bing Jiang, Cai Ju, Billy Jun, Patrick LeGresley, Libby Lin, Junjie Liu, Yang Liu, Weigao Li, Xiangang Li, Dongpeng Ma, Sharan Narang, Andrew Ng, Sherjil Ozair, Yiping Peng, Ryan Prenger, Sheng Qian, Zongfeng Quan, Jonathan Raiman, Vinay Rao, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Kavya Srinet, Anuroop Sriram, Haiyuan Tang, Liliang Tang, Chong Wang, Jidong Wang, Kaifu Wang, Yi Wang, Zhijian Wang, Zhiqian Wang, Shuang Wu, Likai Wei, Bo Xiao, Wen Xie, Yan Xie, Dani Yogatama, Bin Yuan, Jun Zhan, and Zhenyao Zhu. 2016. Deep speech 2 : End-to-end speech recognition in English and Mandarin. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 173–182, New York, New York, USA. PMLR.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Nora Barroso, Odei Barroso, Unai Susperregi, Aitzol Ezeiza, and Karmele López de Ipiña. 2010. Language identification oriented to multilingual speech recognition in the Basque context. In *2010 IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA 2010)*, pages 1–8.

Suvarnsing G. Bhable and Charansing N. Kayte. 2020. Review: Multilingual acoustic modeling of automatic speech recognition (ASR) for low resource languages. In *2020 IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI)*, pages 1–4.

Rafal Černiavski. 2022. Cross-lingual and multilingual automatic speech recognition for Scandinavian languages. Master's thesis, Uppsala University.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2021. Unsupervised Cross-Lingual Representation Learning for Speech Recognition. In *Proc. Interspeech 2021*, pages 2426–2430.

Lars-Olof Delsing and Katarina Lundin Åkesson. 2005. *Håller språket ihop Norden?: en forskningsrapport om ungdomars förståelse av danska, svenska och norska [Does the Language Keep the Nordic Countries Together? A Research Report on How Well Young People Understand Danish, Swedish and Norwegian]*. Nordic Council of Ministers, Copenhagen, Denmark.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. Deep speech: Scaling up end-to-end speech recognition.

Alexander Håkansson and Kevin Hoogendijk. 2020. Transfer learning for domain specific automatic speech recognition in Swedish: An end-to-end approach using Mozilla's DeepSpeech. Master's thesis, Chalmers tekniska högskola / Institutionen för data och informationsteknik, Gothenburg, Sweden.

Hui Lin, Li Deng, Dong Yu, Yi-fan Gong, Alex Acero, and Chin-Hui Lee. 2009. A study on multilingual acoustic modeling for large vocabulary ASR. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4333–4336.

Dau-Cheng Lyu and Ren-Yuan Lyu. 2008. Language identification on code-switching utterances using multiple cues. In *INTERSPEECH 2008, 9th Annual Conference of the International Speech Communication Association, Brisbane, Australia, September 22-26, 2008*, pages 711–714.

Koena Ronny Mabokela and Madimetja Jonas Manamela. 2013. An integrated language identification for code- switched speech using decoded-phonemes and support vector machine. In *2013 7th Conference on Speech Technology and Human - Computer Dialogue (SpeD)*, pages 1–6.

Martin Malmsten, Chris Haffenden, and Love Borjeson. 2022. Hearing voices at the National Library - a speech corpus and acoustic model for the Swedish language. *ArXiv*, abs/2205.03026.

Vineel Pratap, Anuroop Sriram, Paden Tomasello, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert. 2020. Massively multilingual ASR: 50 languages, 1 model, 1 billion parameters. In *Proceedings of Interspeech 2020*, pages 4751–4755.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.

Magnus Sahlgren, Fredrik Carlsson, Fredrik Olsson, and Love Börjeson. 2021. It's basically the same language anyway: the case for a Nordic language model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 367–372, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018. 82 treebanks, 34 models: Universal Dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Jörg Tiedemann. 2009. Character-based PSMT for closely related languages. In *Proceedings of the 13th Annual conference of the European Association for Machine Translation*, Barcelona, Spain. European Association for Machine Translation.

Dong Yu and Li Deng. 2015. *Automatic speech recognition: A Deep Learning Approach*. Springer-Verlag, London.

A character-based analysis of impacts of dialects on end-to-end Norwegian ASR

Phoebe Parsons[1] **Knut Kvale**[2] **Torbjørn Svendsen**[1] **Giampiero Salvi**[1,3]

[1]Department of Electronic Systems, NTNU, Trondheim, Norway
[2]Telenor Research, Oslo, Norway
[3]KTH Royal Institute of Technology, EECS, Stockholm, Sweden
{phoebe.parsons, torbjorn.svendsen, giampiero.salvi}@ntnu.no, knut.kvale@telenor.com

Abstract

We present a method for analyzing character errors for use with character-based, end-to-end ASR systems, as used herein for investigating dialectal speech. As end-to-end systems are able to produce novel spellings, there exists a possibility that the spelling variants produced by these systems can capture phonological information beyond the intended target word. We therefore first introduce a way of guaranteeing that similar words and characters are paired during alignment, thus ensuring that any resulting analysis of character errors is founded on sound substitutions. Then, from such a careful character alignment, we find trends in system-generated spellings that align with known phonological features of Norwegian dialects, in particular, "r" and "l" confusability and voiceless stop lenition. Through this analysis, we demonstrate that cues from acoustic dialectal features can influence the output of an end-to-end ASR systems.

1 Introduction

Automatic Speech Recognition (ASR) has, like all machine learning tasks, struggled with generalization. That is, a model will perform well on the task and data it was trained on but when presented with new examples, especially examples that differ in some dimension from the training data, the model will perform markedly less well. In the task of ASR, this means that models often struggle with generating correct transcriptions for speakers whose age, gender, or dialect differs from that of the speakers on which the model was originally trained. Of specific focus in this paper is the impact of dialect on a modern ASR system.

Dialect information has been used in different ways in ASR. In some applications, such as Dialect Identification (DID), the goal is to correctly identify the dialect for a given sample of speech. Hämäläinen et al. (2021), for example, used a combination of speech and text features to perform DID. In other cases, DID is combined with ASR systems to improve transcription accuracy. For example, Zhang and Hansen (2018) used bottleneck features extracted via unsupervised deep learning to perform DID for both Chinese and Arabic. Similarly, Imaizumi et al. (2022) used a multitask model for both DID and ASR. This multitask approach outperformed the single task systems on both DID and ASR.

Beyond DID, the behavior of ASR systems has been analyzed with respect to dialectal speech (as we do in this paper). This in order to explore phonetic phenomena, as well as to gain insights into the way those complex systems work. In these studies, even when dialectal information is not an explicit target, there is still an interest to understand what phonetic and dialectal information has been captured in ASR models. With traditional ASR models, this investigation has been fairly straightforward as these models have consisted of three semi-independent components: the acoustic model, the language model, and the lexicon. Because of the separate acoustic models within these multi-component models, one could, for example, perform clustering on the model parameters themselves such as (Salvi, 2003a,b, 2005). In this work, Salvi performed clustering on the acoustic model features and correlated the resulting clusters with known dialectally realized phonemes. Instead of directly using an acoustic model from an ASR system, Chen et al. (2014) adapted the concept of an HMM acoustic model to automatically discover dialect-specific phonetic rules.

Unlike multi-component ASR systems, investigating modern, end-to-end models for phonetic and dialectal information is quite different.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 467–476
May 22-24, 2023

Whereas parameters from an acoustic model may be extracted and used independently, the acoustic information in an end-to-end model cannot be so easily excised. This design makes it more challenging, but not impossible, to investigate what acoustic information is captured where in the network. Belinkov and collaborators used the output from each layer of an end-to-end system to train phonetic, grapheme, and articulatory classifiers (Belinkov and Glass, 2017; Belinkov et al., 2019). Prasad and Jyothi (2020) investigated dialectal information captured by an end-to-end system using not only layer-wise classification but also gradient and information-theoric analysis. All of these works are focused on analyzing the network-internal representations detached from actual network output.

The output from ASR models is constrained by the model architecture. Traditional ASR models with lexicons are bound to output only words contained within that lexicon. This means that all transcripts generated by these models contained only real, known words even if the transcribed output did not necessarily match the word that was spoken. Additionally, these models do not allow for acceptable variation in spelling. For example, the word, "favorite," would always be spelled "favorite" never "favourite," even if the latter might better reflect the preference of a British English speaker. Conversely, these newer end-to-end architectures, trained using connectionist temporal classification (CTC) loss, produce output at the character instead of word level. This permits the model to create novel words and spellings, potentially better reflecting the phonetic realization of the spoken word. Given that CTC models are allowed to generate novel spellings, there exists the potential that dialectal information will be captured by the model output itself via non-standard spellings.

The goal of this paper is to investigate whether dialectal acoustic information can impact spellings with an end-to-end model. In order to test this, we used wav2vec 2.0 (Baevski et al., 2020) to generate transcriptions of Norwegian speech. We then performed an analysis of the resulting transcripts for captured dialectal knowledge via a dialectal-region based evaluation of character error patterns. From this analysis we are able to see known Norwegian dialectally-based phonological patterns, specifically around "r" and "l" confusability and stop consonant voicing. Thus we illustrate that strong enough acoustic dialectal cues can effect the character output of an end-to-end ASR system.

2 Norwegian language and dialects

In this paper, we focus our analysis on the Norwegian language. Though spoken by a relatively small population of a little over 5 million speakers, Norwegian contains many dialects differentiated in phonology, syntax, and lexicon. In addition to dialectal variation, Norwegian also maintains two official written standards: Bokmål and Nynorsk; though neither written standard directly corresponds with a spoken variant. Furthermore, Norway does not recognize any official language standard. Indeed, people are encouraged to use their preferred written standard and native dialect in all aspects of work and life.

The variety in dialects stems from Norway's challenging and rugged topography that has historically forced the populace to organize into many, smaller communities. Over time, the diversity we see in Norwegian dialects developed in these small, isolated communities. As described by phoneticians, there now exist large dialectal phonetic variations ranging from infinitive verb endings to palatalization of consonants, to /r/ and /l/ realizations, to the various pronunciations for the personal pronoun for "I", *jeg* —ranging from [jæi] to [eg] to [i] and more (Skjekkeland, 1997).

While the number of specific Norwegian dialects is quite large, we can group these dialects into larger dialect groups for the purpose of this investigation. These grouping could be either into the regional names used by Skjekkeland or into the even larger, cardinal regions of "East," "West," "North," "South," and "Mid." The analysis outlined in this paper relies on these cardinal regions.

3 Methods

3.1 Experimental setup and data

In order to investigate the impact of dialect on an end-to-end ASR system, a well-performing baseline model was required. Therefore, we used three models trained by the Norwegian National Library AI Lab and released publicly on the Hugging Face

repository for our analysis [1][2][3]. The first model contained one billion parameters and was originally trained on the XLS-R (Babu et al., 2021). It was then fine tuned using the Norwegian Parliamentary Speech Corpus (NPSC) to transcribe Norwegian Bokmål text. The other two models were fine tuned from the 300 million parameter VoxRex model (Malmsten et al., 2022). One of these 300 million parameter models was fine-tuned to transcribe Bokmål, the other Nynorsk. All models use a 5-gram word-based language model. In all cases, the NPSC corpus was used to fine-tune the models (Solberg and Ortiz, 2022). When evaluated against the NPSC corpus, the Norwegian AI lab reports a word error rate (WER) of 6.33% for the 1 billion parameter model, 7.03% for the 300 million parameter Bokmål model, and 12.22% for the Nynorsk model. These results indicate that these models will make excellent candidates for our analysis.

As stated earlier, the models to be used were trained on the NPSC. This consists of recordings from the Norwegian Parliament and thus the speech style can be considered mostly spontaneous, with perhaps slightly more planning than everyday speech. For analysis purposes, the NPSC was excluded. This is due to data sparsity in the NPSC test set. While the whole test set is acceptable for model evaluation, data becomes untenably sparse when considered dialect-by-dialect. Thus our analysis focuses on results from two unrelated and more dialectally robust corpora: Rundkast and NB Tale.

The Rundkast corpus consists of radio broadcasts from the Norwegian Broadcasting Corporation (NRK) (Amdal et al., 2008). These transcripts are in both Bokmål and Nynorsk which are treated separately for analysis in this paper. Dialectal annotations were added by the transcribers during corpus creation and are provided directly in the speaker metadata.

NB Tale is publicly available from the National Library of Norway's Language Bank and consist of recordings and transcripts of native and non-native speakers of Norwegian. All speech was transcribed using the Bokmål standard. Read speech was recorded from both the native and non-native speakers whereas spontaneous speech was only recorded for the native speakers. For the analysis in this paper only speech from the native speakers was used. For each speaker biographical information was collected, including the municipality in Norway where they lived as a child. From this municipality, a manual mapping to dialect was devised. This mapping then allowed us to infer the speaker's most likely dialect.

Data was prepared and standardized according to the scripts provided in the combined data set, as described by (Solberg et al., 2023). This converted all audio to a mono, 16kHz format. The text was normalized such that capitalizations, punctuation, and hesitations were removed. Additionally, all non-standard forms were converted into a standard equivalent.

3.2 Word and character alignment

As our investigation into dialectal impact revolves around analyzing trends in character errors, we require an alignment between reference text and model-generated hypothesis text where words that only differ by a few characters are prioritized for alignment. While character error rate (CER) computed across a whole utterance is useful in understanding an aggregate of character errors, this method loses awareness of word boundaries. For example, "også kalt" and "og såkalt" would be aligned in whole-utterance CER with an insertion and a deletion of a space (resulting in "og så kalt"). However, we prefer an alignment where we recognize that "så" was removed from the first word and "så" as added to the second word. Thus CER, as it is generally used across entire utterances, does not answer for our analysis purposes.

With traditional, word-level Levenshtein-based alignments, word similarity is not considered. Any pair of words that do not exactly match are treated as completely different. However, by considering word similarity, the resulting alignments can be used for analysis of broad trends of spellings (e.g., a word ending in "a" instead of "e") that can indicate dialectal impact.

To accomplish such an alignment, an extension to the traditional Levenshtein alignment was developed (Levenshtein, 1965). Typically edit costs are fixed at a value before alignment is computed. However, in our solution instead of a fixed cost for substitutions, we allow it to be dynamically

[1] https://huggingface.co/NbAiLab/nb-wav2vec2-1b-bokmaal

[2] https://huggingface.co/NbAiLab/nb-wav2vec2-300m-bokmaal

[3] https://huggingface.co/NbAiLab/nb-wav2vec2-300m-nynorsk

computed as the CER between the two candidate words. This still ensures that there is no cost for aligning words that are the same while also preferring substitutions of similarly spelled words.

	voiced	class	nasal	place	rounding
"k"	0	0	0	5	0
"g"	1	0	0	5	0
"n"	1	0	1	2	0

Table 1: Example of the vectors for "k", "g", and "n" for Norwegian. Indexes of the vector represent features and values represent their realization.

	height	front	rounding
"a"	2	0	0
"e"	1	2	0

Table 2: Example of the vectors for "a", and "e" for Norwegian. Indexes of the vector represent features and values represent their realization.

Once word-level alignment is computed using the dynamic substitution cost, we can investigate spelling errors. To ensure characters within a word are aligned optimally, we continue to use the dynamic substitution cost idea and compute the substitution cost between characters as the Euclidean distance between two feature vectors. To support this, articulatory feature vectors were created for each letter in the Norwegian alphabet using the International Phonetic Alphabet (IPA) charts as a guide. Articulatory features were considered as indexes in the vector and the values correspond to the realization. For our work, consonants (see examples in Table 1) were defined and treated separately from vowels (see examples in Table 2). As the goal with these vectors is not to create an accurate grapheme-to-phoneme mapping, nor to perfectly illustrate all possible IPA nuance, but instead to align letters in a more logical way, these vectors were sufficient.

To illustrate the necessity of these vectors, consider the word pair of *inngang* (meaning "entrance") and *enkel* ("easy"). Using a traditional alignment method [4], where all characters substitutions have the same cost, an alignment like in

[4]Alignment generated using the Python Levenshtein package: `https://github.com/maxbachmann/python-Levenshtein`

reference	i	n	n	g	a	n	g
hypothesis	e	n			k	e	l

Table 3: A possible alignment between *inngang* and *enkel*, generated without accounting for character similarity.

reference	i	n	n	g	a	n	g
hypothesis	e	n		k	e	l	

Table 4: A possible alignment between *inngang* and *enkel*, generated by accounting for character similarity.

Table 3 is generated. However, using articulatory features as a distance, we are able to generate the alignment in Table 4 where "g" and "k" (only differing by voicing), "a" and "e" (both being front vowels), and "n" and "l" (both being sonorants) are aligned.

While this solution is slightly phonologically flawed —wholly ignoring the di- and trigraphs that exist in Norwegian and instead treating the component letters individually, for example —these feature vectors do accomplish the goal of creating a logical character-level alignment. With confidence in our word and character alignment we can perform the investigation into character substitution trends that constitutes our results.

4 Results

4.1 WER by dialect

To first understand the general trend in recognition across dialects, the WER was calculated for each dialect across the whole of the Rundkast and NB Tale corpora. Transcriptions were generated using both the 300 million and 1 billion parameter Bokmål models for both corpora. Rundkast was further transcribed with the 300 million parameter Nynorsk model (since Rundkast actually contains Nynorsk utterances, unlike NB Tale).

As displayed in Table 5 that shows WER across both corpora and dialects, we can see WER values ranging from the low teens to nearly 40%. These values are markedly higher than the 6.33% WER that was reported on the NPSC which highlights the impact of domain mismatch on ASR; models trained on one domain (the Norwegian Parliament) do not generalize well to new domains (radio and studio recordings).

Dataset	Dialect	Utterances	Speakers	WER%		
				1B Bok	300M Bok	300M Ny
NB Tale —Bokmål utterances	Other	5087	120	25.79	26.04	—
	West	4064	93	20.34	20.78	—
	Mid	1789	40	18.14	20.02	—
	North	2760	68	17.89	18.54	—
	South	591	14	16.86	18.00	—
	East	1898	42	16.44	17.15	—
Rundkast —Bokmål utterances	Unknown	199	12	19.54	18.28	38.82
	West	7526	176	18.21	16.66	36.28
	Mid	2917	124	17.06	17.35	37.30
	North	2941	153	16.38	16.13	35.31
	South	1372	56	16.16	15.11	35.67
	East	51303	993	13.93	13.35	36.04
Rundkast —Nynorsk utterances	South	355	15	31.63	30.46	31.89
	Mid	77	1	30.41	29.46	27.89
	West	6024	161	29.35	28.26	23.99
	East	2802	34	28.27	26.96	20.49
	North	13	1	26.43	27.86	18.12
	Unknown	3	3	0.00	0.00	0.00

Table 5: WER for Rundkast and NB Tale corpora. Transcribed using the all models. As there is no Nynorsk text in the NB Tale corpus, we did not evaluate the Nynorsk model. The WER reported for the models on the NPSC corpus are 6.33% for the 1B model, 7.03% for the 300M Bokmål model, and 12.22% for the 300M Nynorsk model.

For the Bokmål text in both corpora, we can see that models perform best on the "East" dialect region whereas the "West" region has the worst performance. It is unclear which model is generally the best. The 1 billion parameter model performs better than the 300 million parameter model on the NB Tale text, but the 300 million parameter model outperforms the 1 billion on the Rundkast text.

With the Rundkast corpus, we can see that the Bokmål models perform, as expected, poorly on the Nynorsk text with the converse (Nynorsk model evaluated against Bokmål text) being true as well. However, even when the Nynorsk model is evaluated against Nynorsk text, the results are still worse than the Bokmål model of the same size evaluated against Bokmål text.

Of more concern than model accuracy, however, is data scarcity for Nynorsk text. Given that Nynorsk is primarily used in the western part of Norway, the nearly equal split of speakers between Bokmål and Nynorsk for the "West" region is understandable. Moreover, for the other regions ("North" and "Mid" in particular) there are too few speakers to draw conclusions from. Therefore, as we move forward with the character-based analysis, we will be focusing on the Bokmål models and their performance on the Bokmål text.

4.2 /r/ and /l/ confusiblity

In Norwegian, /r/ is generally realized as either a voiced apical tap or a voiced velar approximant (Kvale and Foldvik, 1992). These two different pronunciations are considered dialect features, with the approximant version predominating in the "South" and "West" of the country and the tap being the norm in the rest of country. The maps in (Kvale and Foldvik, 1999) and (Skjekkeland, 1997) nicely illustrate this distribution.

Similar to the Norwegian /r/, which can be realized in several variants, the Norwegian /l/ also has dialectally motivated realizations. Many speakers in the "East", "Mid", and southern part of the "North" region of the country produce a voiced retroflex flap. The norm for speakers in the rest of the country ("West", "South", and the remaining part of the "North") is a voiced dental/alveolar lateral (Kvale and Foldvik, 1995).

Understanding these phonetic realizations, we can anticipate that the tapped [ɾ] and the lateral approximant [l] should be minimally confusing for

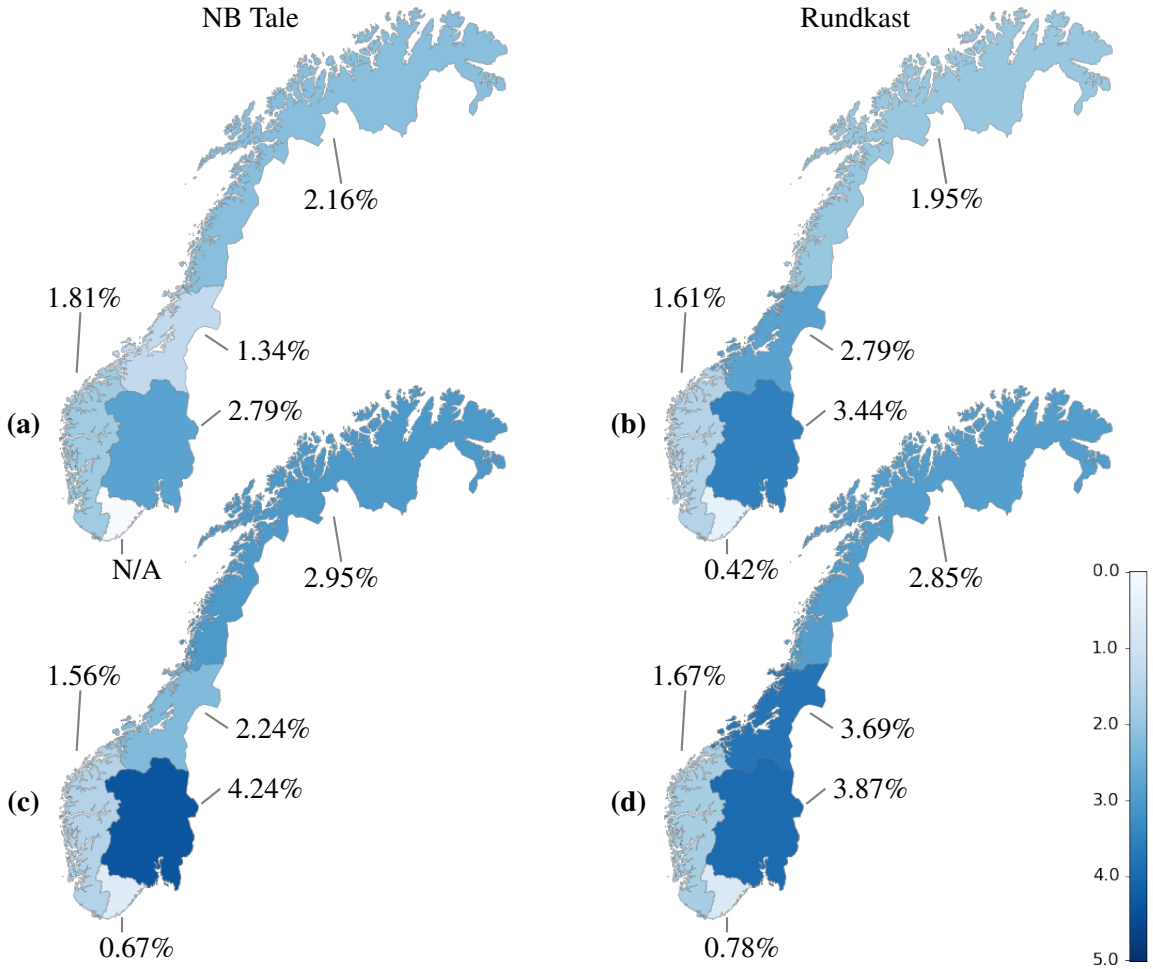


Figure 1: Instances of "r" becoming "l". The first column (a, c) show results on the NB Tale utterances; second column (b, d) shows results on the Rundkast utterances.The first row (a, b) being results from the 300m model and the second row (c, d) being results from the 1b parameter model.

the model. The former being a brief interruption in the airflow and the latter being a continuous, smooth approximant. However, for speakers in the "East" and "Mid" parts of the country, where both the tapped [ɾ] and flapped [ɽ] dialect features are present, we would anticipate a greater degree of confusion. Both tapped [ɾ] and flapped [ɽ] are seen as brief closures with acoustic differentiation relegated to the F3 and F4 trajectories (Kvale and Foldvik, 1995).

Therefore to evaluate how much of an impact these potentially similar realizations have on the model, we used the aligned Bokmål texts (as described in Section 3.2) and calculated how frequently "r" was transcribed instead of "l" and vice versa. When analyzing instances of "r" transforming into "l", we only considered instances where the "r" did not precede another alveolar consonant ("t", "d", "n", "l", "s"). This is due to the fact that "r", when followed by an alveolar consonant, can be interpreted as a digraph. In dialect regions with the alveolar [ɾ], speakers will realize the second alveolar consonant as a retroflex instead of pronouncing two distinct sounds. That is, "rt" would be realized as [ʈ]). To ensure these realizations did not cloud our analysis, we excluded all "r"s followed by an alveolar consonant.

The maps in Figures 1 and 2 show the percentage of error. That is, for those instances where an "r" was not transcribed correctly, the maps show what percentage of those errors were because an "l" was transcribed instead (Figure 1). And vice versa for the "l" to "r" transformation (Figure 2). This error calculation and plotting was done for each of the cardinal dialect region. Darker colors represent higher errors. In both figures the first column (a, c) show results on the NB Tale utterances; second column (b, d) shows results on the

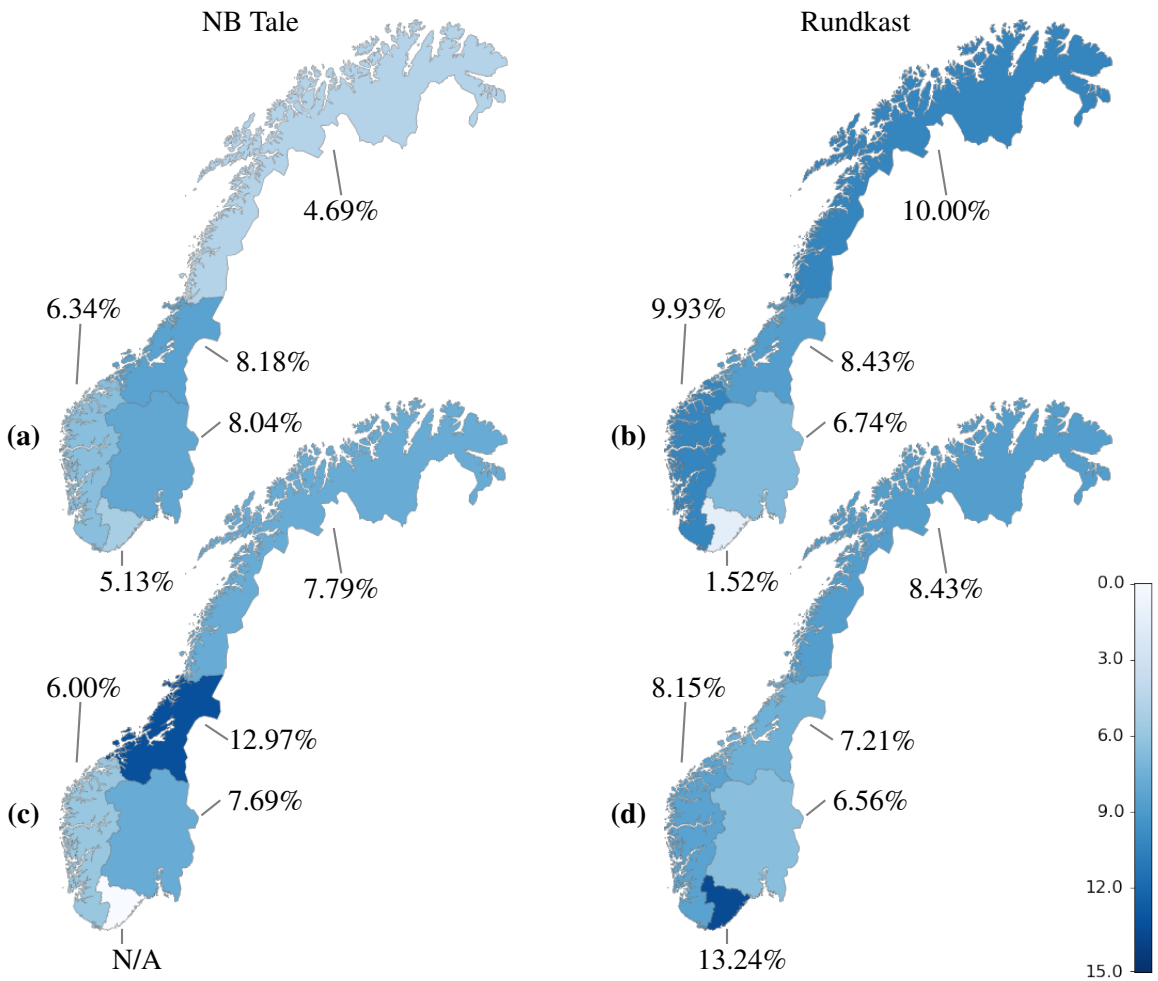


Figure 2: Instances of "l" becoming "r". The first column (a, c) show results on the NB Tale utterances; second column (b, d) shows results on the Rundkast utterances.The first row (a, b) being results from the 300m model and the second row (c, d) being results from the 1b parameter model.

Rundkast utterances. The first row (a, b) being results from the 300m model and the second row (c, d) being results from the 1b parameter model.

For all Figures, except 2(b) and 2(d), the regions with the most confusability between "r" and "l" are the "East", "Mid", and "North". Indeed, for all Figures except 2(d) the "South" has the lowest incidences of "r" and "l" confusion. By and large we also see much clearer, more consistent trends with the NB Tale data. This could be because the utterances in the NB Tale corpora were selected for phonological coverage and thus there were more environments for "r" and "l" confusion.

4.3 Voiceless stop lenition

In addition to /r/ and /l/ confusability, we also investigated the distribution of voiceless stop consonants. In the "South" region, voiceless stops tend to lenite to their voiced counterparts in post-vocalic environments (Skjekkeland, 1997). Thus, we would expect [p], [t], and [k] to lenite to [b], [d], and [g] when preceded by a vowel. To understand if this change is captured by the wav2vec model, we found instances where a voiceless stop was changed and then ensured that the change was to its voiced counterpart. If a voiceless to voice change occurred, we then ensured that both the voice and voiceless stops were preceded by a vowel. We counted occurrences of this post-vocalic voicing change across all three stops of interest. Results can be see in Figure 3 for the NB Tale data and Figure 4 for Rundkast. The first column (a) shows results from the 300m parameter model, second column (b) shows results from 1b parameter model. Darker colors represent higher errors.

For both the NB Tale and Rundkast corpora we,

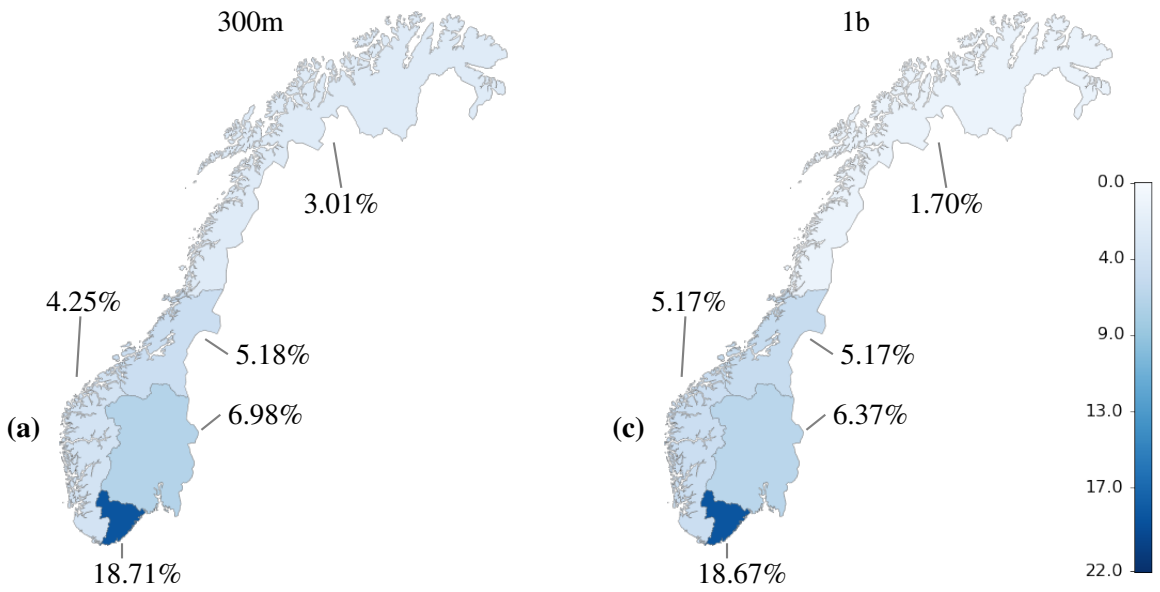


Figure 3: Percentage of postvocalic voicing error; that is, instances of (“p”, “t”, “k”) realized as (“b”, “d”, “g”) as a percentage of total (“p”, “t”, “k”) errors on the NB Tale dataset. First column (a) shows results from the 300m parameter model, second column (b) from the 1b parameter model

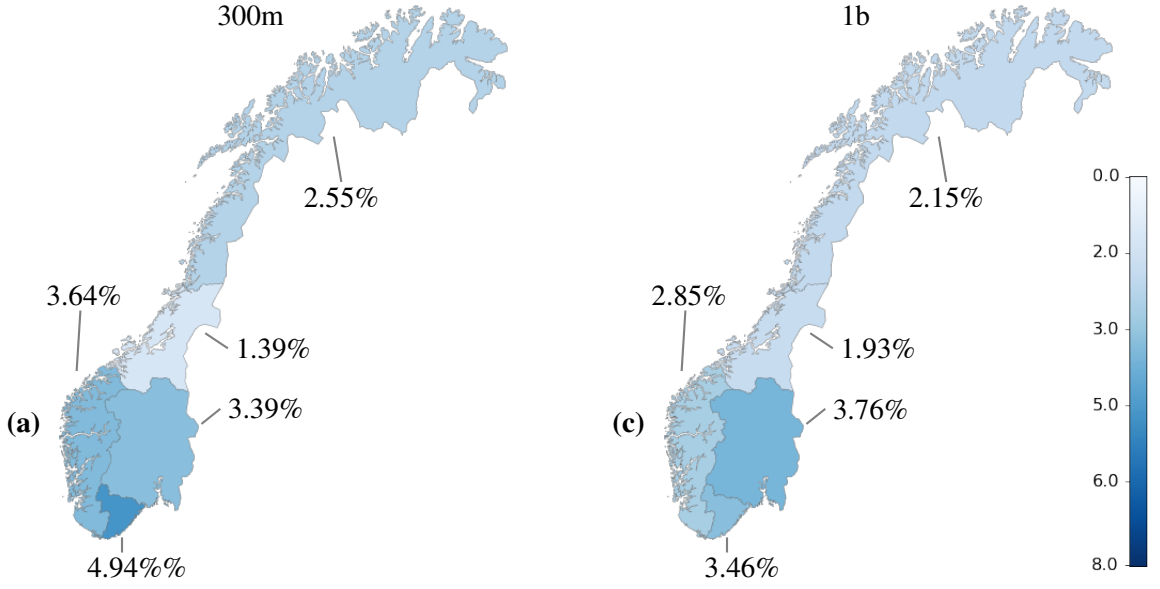


Figure 4: Percentage of postvocalic voicing error; that is, instances of (“p”, “t”, “k”) realized as (“b”, “d”, “g”) as a percentage of total (“p”, “t”, “k”) errors on the Rundkast dataset. First column (a) shows results from the 300m parameter model, second column (b) from the 1b parameter model

can see that the "South" region has the highest instances of voicing. Though once again, we see stronger trends in the NB Tale data then in Rundkast.

4.4 Personal pronoun *jeg*

As mentioned when discussing the Norwegian language in Section 2, there are many ways for Norwegain speakers to say the first person pronoun *jeg*. This was briefly investigated as well. Confusion pairs for *jeg* were aggregated and trends sought. Regardless, no trends in the words substituted for *jeg* in the transcripts could be found. This lack of results could indicate that a word like *jeg* occurs so frequently in all dialects that there is an abundance of training examples for the model to generalize from. Or, perhaps, the 5-gram language model used, in addition to the wav2vec component, had enough influence to ensure that only *jeg* was produced.

5 Discussion

Due to the fact that we have been able to largely see acoustic dialectal features surfacing through our analysis, we find that this method of carefully aligning text and aggregating results has promise. Furthermore, we infer that the models have learned enough about Norwegian to understand standard spellings and apply these generalizations to broader contexts. Additionally, the phonetic information in the dialects is strong enough to cause the models to utilize this general spelling knowledge and create more acoustically aligned outputs. However, going so far as to say that the models have internalized some knowledge about the dialects themselves (e.g., phonetic features) is perhaps more than can be reasonably asserted from this analysis.

Through this paper we have explored a couple of known dialectally-motivated phonological realizations. There still, however, exist more that could be explored. As mentioned in Section 4.2, there exists a pattern of retroflexting of alveolar consonants for certain Norwegian dialects. This analysis could certainly be extended to those environments. However, there are also phonological changes that are hard, or potentially impossible to see in spelling changes. For example, alveolars are palatalized (most strongly) in the "Mid" region as well as in certain phonological environments in the "North" and the northern parts of the "West" and "East" regions. This palatalization would be hard to see in spellings since there is no standard way in Norwegian orthography of representing a palatalized sound. Additional Norwegian phonological features that have no written representation (such as toneme) would also be invisible to the analysis performed in this paper.

As the NPSC is derived from parliamentary speeches, the distribution of parliament speakers emulates the population distribution of the country. Thus our models, all of which were trained on NPSC, have the same speaker representation. That is, the "East" region would be the most represented in the training data. Given this, and the results in Table 5, it would seem that the models have best learnt the features which they saw the most, as machine learning models are wont to do. Therefore, if models are to be robust against dialects, it seems necessary to increase the training data for the other regions. Additionally, it might be possible to assign greater weight to these dialectal character changes during training to encourage the models to learn a better representation.

6 Conclusion

Through this paper, we demonstrate how an analysis of character errors in transcriptions generated by an end-to-end ASR system can contain dialectal trends mirroring those known through linguistic descriptions. We showed increased confusability between "r" and "l" in regions where those phonemes are realized similarly. We also showed increased incidences of voiceless stop lenition in a region known for that phenomena. These errors indicate that the end-to-end system has successfully learnt to spell in Norwegian, going so far so as to slightly spell in dialect.

7 Acknowledgements

This work has been done as part of the SCRIBE project as funded by the Norwegian Research Council, project number: 322964.

References

Ingunn Amdal, Ole Strand, Jørn Almberg, and Torbjørn Svendsen. 2008. Rundkast: an annotated norwegian broadcast news speech corpus. pages 1907–1913.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan

Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. Xls-r: Self-supervised cross-lingual speech representation learning at scale. *arXiv*, abs/2111.09296.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Yonatan Belinkov, Ahmed Ali, and James Glass. 2019. Analyzing Phonetic and Graphemic Representations in End-to-End Automatic Speech Recognition. In *Proc. Interspeech 2019*, pages 81–85.

Yonatan Belinkov and James Glass. 2017. Analyzing hidden representations in end-to-end automatic speech recognition systems. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 2438–2448, Red Hook, NY, USA. Curran Associates Inc.

Nancy F. Chen, Sharon W. Tam, Wade Shen, and Joseph P. Campbell. 2014. Characterizing phonetic transformations and acoustic differences across english dialects. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):110–124.

Mika Hämäläinen, Khalid Alnajjar, Niko Partanen, and Jack Rueter. 2021. Finnish dialect identification: The effect of audio and text. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8777–8783, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ryo Imaizumi, Ryo Masumura, Sayaka Shiota, and Hitoshi Kiya. 2022. End-to-end japanese multi-dialect speech recognition and dialect identification with multi-task learning. *APSIPA Transactions on Signal and Information Processing*, 11.

Knut Kvale and Arne Kjell Foldvik. 1992. The multifarious r-sound. In *Proc. International Conference on Spoken Language Processing (ICSLP-92)*, pages 1259–1262.

Knut Kvale and Arne Kjell Foldvik. 1995. An acoustic analysis of the retroflex flap. In *Proc. International Congress of Phonetic Sciences (ICPhS-95)*, pages 454–457.

Knut Kvale and Arne Kjell Foldvik. 1999. Mapping dialect characteristics to dialect speakers. In *ICPhS-14*, pages 1613–1616.

Vladimir I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710.

Martin Malmsten, Chris Haffenden, and Love Börjeson. 2022. Hearing voices at the National Library – a speech corpus and acoustic model for the Swedish language. *arXiv*.

Archiki Prasad and Preethi Jyothi. 2020. How accents confound: Probing for accent information in end-to-end speech recognition systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3739–3753. Association for Computational Linguistics.

Giampiero Salvi. 2003a. Accent clustering in Swedish using the Bhattacharyya distance. In *ICPhS-15*, pages 1149–1152.

Giampiero Salvi. 2003b. Using accent information in ASR models for Swedish. In *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, pages 2677–2680.

Giampiero Salvi. 2005. Advances in regional accent clustering in Swedish. In *Proc. Interspeech 2005*, pages 2841–2844.

Martin Skjekkeland. 1997. *Dei norske dialektane : tradisjonelle særdrag i jamføring med skriftmåla*. Høyskoleforl.

Per Erik Solberg and Pablo Ortiz. 2022. The Norwegian parliamentary speech corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1003–1008, Marseille, France. European Language Resources Association.

Per Erik Solberg, Pablo Ortiz, Phoebe Parsons, Torbjørn Svendsen, and Giampiero Salvi. 2023. Improving generalization of Norwegian ASR with limited linguistic resources. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*. Linköping University Electronic Press, Sweden.

Qian Zhang and John H. L. Hansen. 2018. Language/dialect recognition based on unsupervised deep learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(5):873–882.

Quasi: a synthetic Question-Answering dataset in Swedish using GPT-3 and zero-shot learning

Dmytro Kalpakchi
Division of Speech, Music and Hearing
KTH Royal Institute of Technology
Stockholm, Sweden
dmytroka@kth.se

Johan Boye
Division of Speech, Music and Hearing
KTH Royal Institute of Technology
Stockholm, Sweden
jboye@kth.se

Abstract

This paper describes the creation and evaluation of a synthetic dataset of Swedish multiple-choice questions (MCQs) for reading comprehension using GPT-3. Although GPT-3 is trained mostly on English data, with only 0.11% of Swedish texts in its training material, the model still managed to generate MCQs in Swedish. About 44% of the generated MCQs turned out to be of sufficient quality, i.e. they were grammatically correct and relevant, with exactly one answer alternative being correct and the others being plausible but wrong. We provide a detailed analysis of the errors and shortcomings of the rejected MCQs, as well an analysis of the level of difficulty of the accepted MCQs. In addition to giving insights into GPT-3, the synthetic dataset could be used for training and evaluation of special-purpose MCQ-generating models.

1 Introduction

OpenAI's GPT-3 (Brown et al., 2020) is the current state-of-the-art model for text generation. One of the more impressive properties of this model is the way it can perform natural-language tasks without any labeled examples, with so-called *zero-shot learning*. In the case of GPT-3, zero-shot learning entails that the model receives a prompt describing the task verbally (for example, "Translate from English to Spanish"), some input data for the task (a text in English in this example), and then produces the output (the Spanish translation in the example).

Most research using GPT-3 has focused on English, because the bulk of GPT-3's training data (92.6% of words) is English text. Only 0.11%[1] of the training data is Swedish text. This might sound insignificant at first, but it actually amounts to 220.9 million words, which is quite a sizeable corpus! In addition, Swedish and English are both Germanic languages, so it is possible that some cross-lingual learning has taken place during training. Taking all this into account, we want to test whether GPT-3 would be able to handle tasks in Swedish in a zero-shot fashion. Specifically, the article has the following two goals.

1. Provide a pilot evaluation of GPT-3's ability to generate multiple-choice questions (MCQ) in a zero-shot manner.

2. Create the first synthetic dataset of MCQs, called *Quasi*,[2] for testing reading comprehension of adult language learners of Swedish.

An MCQ consists of a text, a question (called the *stem*) on the text, and a set of answer alternatives, of which exactly one is correct (called the *key*) and all the others are wrong, but plausible (called *distractors*). As we will show, GPT3 is good but far from perfect in generating Swedish MCQs from a Swedish text: more than half of the generated MCQs were incorrect, sometimes in subtle ways. This means that GPT-3 does not provide an ultimate solution to the MCQ-generation task, and that special-purpose models are still required. The synthetic dataset presented here could potentially be used as extra training material for such special-purpose models.

In this paper, we provide a detailed analysis of the errors and shortcomings of the rejected MCQs, as well as an analysis of the level of difficulty of the accepted MCQs, giving insights into the strengths and weaknesses of GPT-3.

[1] Hyperlink to a CSV file with the training data statistics

[2] Raw data, annotations, the details on the annotation setup, and the source code are available at https://github.com/dkalpakchi/Quasi

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 477–491

2 Related work

The idea of creating synthetic datasets is not new both in NLP in general (Gessler et al., 2020; He et al., 2022), and for Question Answering (QA) specifically (Alberti et al., 2019). To the best of our knowledge, no synthetic datasets of multiple choice questions (MCQs) for testing reading comprehension have been created either for English or Swedish. However, for English, the use of synthetic MCQs has been explored for other domains, such as natural sciences (Le Berre et al., 2022) or factual QA (Puri et al., 2020).

QA for Swedish is an under-researched area with very few existing datasets. There has been an attempt to translate SQuAD (Rajpurkar et al., 2016), which does not contain MCQs, into Swedish[3] with no information on whether translations were manually checked. To the best of our knowledge, the only existing MCQ dataset in Swedish is SweQUAD-MC (Kalpakchi and Boye, 2021), which has been manually constructed.

3 Data collection

3.1 Textual materials

We have collected 96 texts of varying length, type and genre from the national tests of Swedish for Immigrants courses (swe. *SFI nationella prov*) using OCR. These texts have been specifically adapted to test reading comprehension of adult language learners of Swedish. The sought-after synthetic data should consist of MCQs for each given text, where each MCQ must fulfill a number of requirements:

1. there must be 4 alternatives;
2. only one alternative must be correct;
3. the other 3 alternatives must be wrong, but plausible;
4. the question must be answerable using the information in the given text.

For each text, a batch of MCQs fulfilling the requirements above should be generated and the number of MCQs in the batch should vary depending on the length of text: the longer the text, the more MCQs should be available. Additionally, the difficulty of MCQs in each batch should vary.

[3] https://github.com/Vottivott/swedsquad

3.2 GPT-3 hyperparameters

We have employed OpenAI's GPT-3 (Brown et al., 2020), more specifically version *text-davinci-003*, to generate synthetic data that fuifils the requirements from the previous section.

3.2.1 Prompt

Our approach to creating the prompt was to spell out the aforementioned requirements as clearly as possible. The results was the following prompt, which has been fed to GPT-3 **in Swedish**.

> Skriv N_q olika läsförståelsefrågor med 4 alternativ (a, b, c, och d) och ge varje fråga en unik nummer (1, 2, 3, osv). Första alternativet (a) ska alltid vara rätt, medan de andra alternativen (b, c, och d) ska vara felaktiga, men troliga. Alla frågor måste kunna besvaras av den följande texten. Ordna frågor från den lättaste till den svåraste.

The number N_q was selected based on the length of each text (the longer the text, the more MCQs we asked for) using the heuristic detailed in Appendix A.

To help non-Swedish-speaking readers, we also provide an English translation of the prompt below, but we emphasize again, that all input to GPT-3, *including* the prompt, was *in Swedish*.

> Write N_q different reading comprehension questions with 4 alternatives (a, b, c, and d) and give each question a unique number (1, 2, 3, and so on). The first alternative (a) should be always correct, while the other alternatives (b, c, and d) should be wrong, but plausible. All questions must be answerable of the following text. Order questions from the easiest to the hardest.

We did **NOT** perform any extensive experimentation with prompt formulation. We have formulated the prompt in a way that it includes all aforementioned requirements in the most unambiguous way possible. Some parts of the requirements are ambiguous by necessity, for instance, the definitions of MCQ difficulty vary among researchers (see Section 4.1 for further discussion on the matter). The intention behind including the difficulty requirement into the prompt was to check whether GPT-3 could produce any variation at all when it comes to MCQ difficulty.

3.2.2 Generation hyperparameters

We did **NOT** perform any systematic search for the generation hyperparameters (e.g., temperature, top P for nucleus sampling, etc). Instead we used the default settings (listed in Appendix B), except for the extended maximum generation length to allow for longer texts and more MCQs.

The rationale behind this decision is that it is impossible to define the degree to which we want GPT-3 to generate repeated content. For instance, if the text consists of one and only sentence: "Stockholm is the capital of Sweden", then one of the few good reading-comprehension questions would be "What is the capital of Sweden?", with the correct answer "Stockholm". In this example, all words from the text are repeated in the question and the correct answer. One could, of course, paraphrase the question to some degree, but then that poses a risk of the question's meaning "drifting away". For instance, the question "What is the administrative center of Sweden?" is still a valid question, but it is neither equivalent to the original question, nor answerable from the given text.

4 Evaluation methodology

We are interested to know how well GPT-3 followed the instructions given in our prompt. For each given text, we investigated the following properties:

Q1. Were N_q MCQs generated?

Q2. Did every MCQ include a stem and 4 alternatives?

Q3. Did the formatting conform to the requested one (MCQs are numbered, alternatives are labeled with letters `a`, `b`, `c`, `d`, etc)?

Q4. Were all MCQs distinct?

The next group of questions is interesting only for distinct MCQs with a stem and 4 alternatives. We will refer to these as *D-questions*, with "D" for "distinct".

D1. Were all stems grammatically correct and answerable after reading the text?

D2. For MCQs having stems compliant with the requirements in D1, were all alternatives grammatically correct and relevant?

The final 3 questions are interesting only for those cases where the answer was *yes* for both D1 and D2. We will refer to these as *R-questions*, with "R" for "relevant".

R1. Was only one alternative always correct, while the others were always wrong, but plausible?

R2. Was the correct alternative always `a`?

R3. Were the MCQs always ordered from the easiest to the hardest?

Although requiring some manual annotation, the questions above are all trivial to check, with the exception of R3, which is non-trivial since the concept of MCQ difficulty is not well-defined. In fact, MCQ difficulty depends on many things that are hard to keep constant, e.g., the reader's skills and background knowledge, whether the test is taken under time pressure, etc. For the purpose of this case study, we have relied on a definition of difficulty outlined in the section below and further detailed in Appendices C.1, C.2, and C.3.

4.1 MCQ difficulty

For defining MCQ difficulty we take inspiration from the methodology proposed by I. Kirsch and P. Mosenthal, which served as one the bases for the TOEFL 2000 (Jamieson et al., 2000) and PISA 2018 (OECD, 2019) reading literacy frameworks. In particular we consulted Kirsch and Mosenthal (1995), because this work specifically deals with assessing difficulty of multiple-choice questions.

Kirsch and Mosenthal (1995) have used the percentage p_c of students who answered the question correctly[4] as a proxy for the MCQ difficulty. In an attempt to explain performance differences, they have defined a number of readability and reading process variables, and ran a regression using these variables as predictors of p_c. They found the following three variables to be particularly strong predictors (later referred to as *core predictors*):

- Type of Information (TOI)
- Type of Match (TOM)
- Plausibility of Distractors (POD)

[4] In Kirsch and Mosenthal (1995), this quantity is called *p-value*, but should not be confused with *p*-values from statistical hypothesis testing, which are also reported using *p*-notation

Inspired by Kirsch and Mosenthal (1995), we evaluated each of the core predictors on a scale from 1 to 5 using the following scoring rules:

- **TOI**: The more abstract the stem, the higher the score. Stems inquiring about concrete things like places or people will get a score of 1, whereas those asking about more abstract concepts will get increasingly higher scores, up to 5 for the most abstract concepts, like themes or patterns.
- **TOM**: The more inference required to match the information in the stem and the key to the text, the higher the score. This means a score of 1 for MCQs requiring simple string matching, up to a score of 5 for those matches requiring reading between the lines.
- **POD**: The closer distractors are to the key in the text, the higher the score. This means a score of 1 for MCQs with no distractors present in the text, up to a score of 5 in the cases where two or more distractors are close to the key in the text.

More precise definitions for scoring the core predictors are provided in Appendix C.1 for Type of Information, Appendix C.2 for Type of Match, and Appendix C.3 for Plausibility of Distractors.

5 Results

Recall that we collected 96 texts and asked GPT-3 to generate N_q MCQs for each of them, where N_q is calculated based on the length of each text (the longer the text, the more MCQs we asked for). In total, GPT-3 made 718 generation attempts. To answer all questions posed in the previous section, we have made all required manual annotations ourselves using an iterative annotation process (annotating – discussing issues – re-annotating). All annotations for this section have been performed using the Textinator[5] annotation tool (Kalpakchi and Boye, 2022).

Q1. Were N_q MCQs always generated?
Answer: No, but very often (for 89.6% of the texts)

For 86 out of 96 texts, GPT-3 generated exactly the requested N_q MCQs. The mismatch between the number of generated MCQs N_{gen} and N_q is shown in Figure 1.

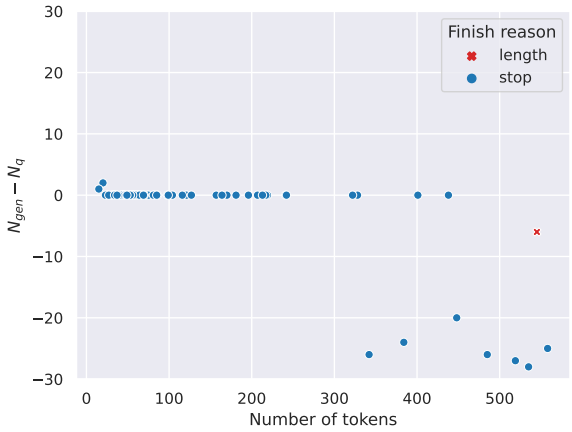


Figure 1: Scatterplot of the relation between the number of tokens (as provided by NLTK) and the size of MCQ number mismatch, $N_{gen} - N_q$

As can be seen, most of the mismatch happens for longer texts and there are mostly fewer MCQs generated than requested. One possible explanation could have been that GPT-3 simply did not have enough tokens in its context window. However, Figure 1 illustrates that in the vast majority of cases, GPT-3 stopped generating MCQs after reaching the stop token. In fact, only in one case was the generation interrupted because the context window was too short (GPT-3 failed even to produce a stem for this example). This means that **717 out of 718** generation attempts resulted in an MCQ.

Q2. Did every MCQ include a stem and 4 alternatives?
Answer: Yes

The only MCQ that did not was the one with the id `0_28`, which was the only failed generation attempt discussed above. All other **717 MCQs** contained a stem with 4 alternatives.

Q3. Did the formatting conform to the requested one (MCQs are numbered, alternatives are labeled with letters `a`, `b`, `c`, `d`, etc)?
Answer: Yes, with some minor variations.

The stems were always numbered using Arabic numbers followed by a full stop. The alternatives were always formatted in the same way both within each MCQ and between all MCQs for each text. The formatting itself has slightly differed between the texts, using either small or capital letters

[5]To facilitate reproducibility, the exact details of the Textinator setup are available in the GitHub repository associated with this paper.

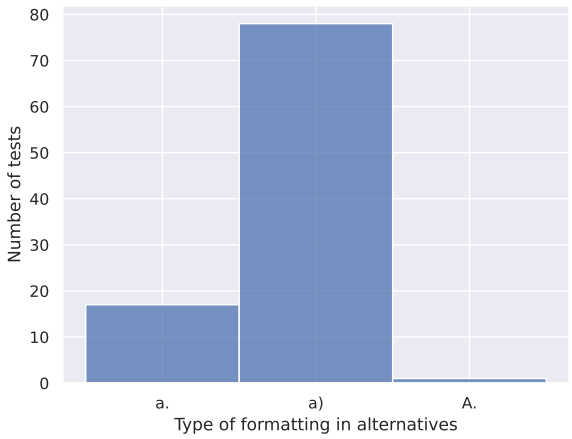


Figure 2: Distribution of the formatting types for the alternatives in each test (using the first alternative `a` as an example). All MCQs within the test were formatted in the same way.

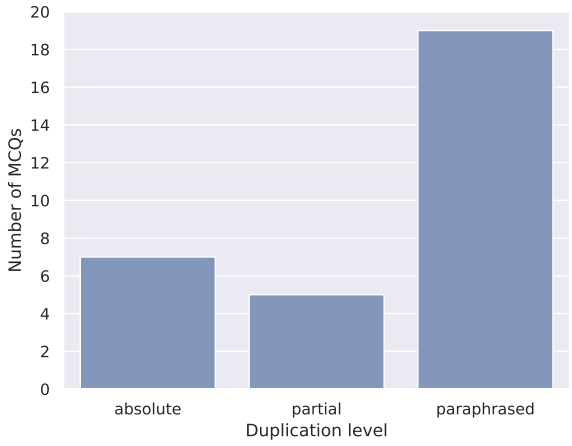


Figure 3: Distribution of duplicated MCQs (4.32% of all MCQs) per duplication level

from `a` to `d`, followed by either a right bracket or a full stop. The distribution of different formatting options is illustrated in Figure 2.

Q4. Were all MCQs distinct?
Answer: Mostly yes (around 4% duplicates).

MCQs can be duplicated to varying extents. We define the following cases, which we call *duplication levels*:

- **absolute** – when both the stem and all alternatives are the same (ignoring the punctuation), **and** the alternatives have been generated in the same order;
- **partial** – when either only the stem is the same, **or** both the stem and all alternatives are the same, but the alternatives have been generated in a different order;
- **paraphrased** – when the stem (and possibly a subset of alternatives) is a paraphrased version of the stem (and possibly a subset of alternatives) of the other MCQ(s).

If one MCQ is a duplicate of more than one MCQ, we take only the strongest duplication level into account. For instance, if X and Y are paraphrased duplicates, whereas X and Z are absolute duplicates, we include X as the case of absolute duplicates in the descriptive statistics.

31 (4.32%) MCQs turned out to be duplicates with a distribution of duplication levels provided in Figure 3. **As previously mentioned, all duplicates are excluded from further analysis.**

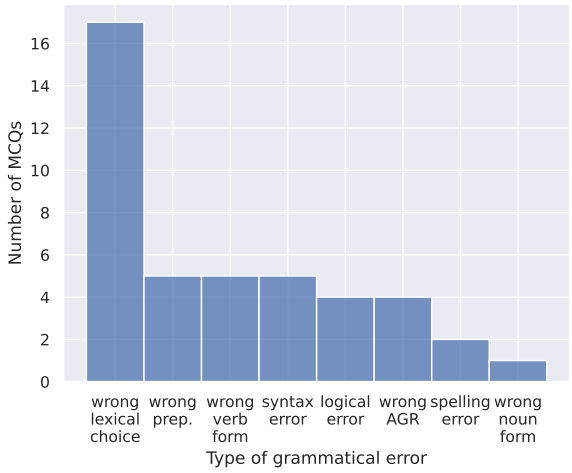


Figure 4: Distribution of grammatical error types for stems. "AGR" stands for "agreement", and "prep." – for "preposition"

D1. Were all stems grammatically correct and answerable after reading the text?
Answer: No (roughly 1 in 5 MCQs did *not* conform to these requirements)

There are multiple kinds of problems related to D1. The first problematic category includes ungrammatical stems, which we have classified further into the types of grammatical errors, shown in Figure 4. In total, **43 (6%) MCQs** had ungrammatical stems with a more detailed description and examples for each grammatical error type given in Appendix D.

The second problematic category concerns the stems that are grammatically correct, but *unanswerable* for the given text. For the purpose of the synthetic data at hand, we have defined the follow-

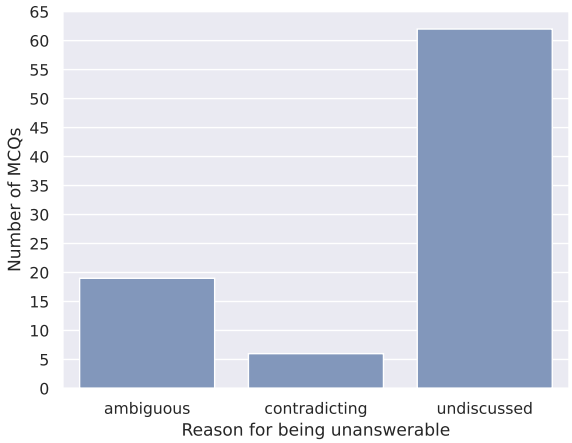


Figure 5: Distribution of reasons for being unanswerable.

ing reasons to classify a stem as unanswerable.

- **Contradictive**, meaning that a presupposition in the stem contradicts what is written in the text. For instance, suppose that in the text it is written "John was very happy to finally resign", and the stem is "Why was John sad about resigning?". Here the presupposition that John was sad is inconsistent with the text. Another example could include the text "John likes playing basketball, but his biggest hobby is tennis" and the stem "What is John's hobby?". This formulation of the stem presupposes that John has one hobby, which is not true and hence inconsistent with the text.

- **Undiscussed**, meaning that the text neither provides the information necessary to find the key for the stem, nor provides the way to reject all but one alternative, while providing some support for the remaining one. In either case the information in the stem does **NOT** contradict the text.

- **Ambiguous**, meaning that the information provided in the stem is not enough to choose one definite answer among the provided alternatives, i.e., different alternative(s) could be viewed as the key, depending on the interpretation of the stem.

87 (12.13%) MCQs were deemed to have unanswerable stems with a more fine-grained distribution depicted in Figure 5.

The last, but not least problematic category in D1 is that of grammatically correct stems that could be answered without reading the text. This category includes **20 (2.79%) MCQs**.

D2. Were all alternatives grammatically correct and relevant for the given stem and text?
Answer: No, but more than for stems (around 3 in 20 MCQs did *not* conform to the requirements above).

Similarly to D1, there are multiple kinds of problems related to D2. One problem is that of ungrammatical alternatives, which uses exactly the same categorization as for D1 (detailed and exemplified in Appendix D) with one additional category: "tautology". In total **10 (1.39%) MCQs** with grammatically correct stems had at least one ungrammatical alternative, with the error type distribution provided in Figure 6.

The other problem concerns cases when the alternatives are grammatically correct, but irrelevant for the given text. For the synthetic data at hand, we have defined the following reasons to judge the alternatives as irrelevant for the given text.

- **Misfocused**, meaning that at least one of the alternatives does not provide the type of information, requested in the stem. One example of such inconsistency would be the stem "What is the capital of Sweden?", accompanied by the alternative "John Lennon". Note that even if the correct answer, "Stockholm", is within the provided 4 alternatives, but so is "John Lennon", the MCQ will still be categorized as misfocused. The rationale is that in

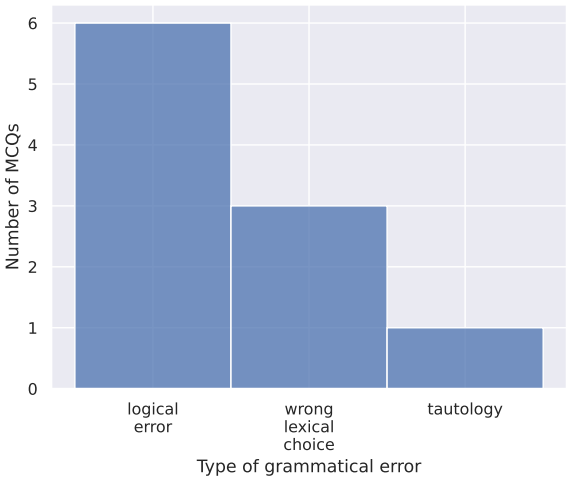


Figure 6: Distribution of grammatical error types for alternatives.

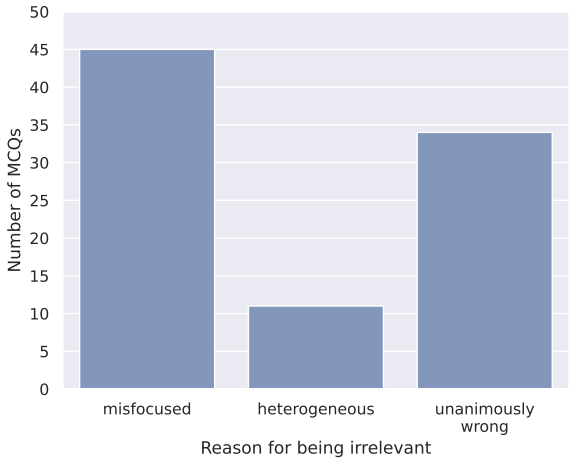


Figure 7: Distribution of reasons for being irrelevant.

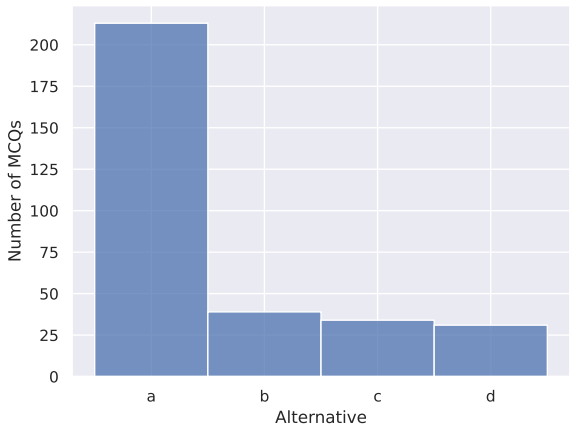


Figure 8: Distribution of the positions of correct alternatives

such cases, the effective number of alternatives becomes less than 4 and thus it becomes easier to guess the correct answer.

- **Heterogeneous**, meaning that one or more of the provided alternatives stick out and thus provide a potential clue for the students. One example would be the stem "Where is Nobel Museum located?" and the alternatives "Stortorget 2, 103 16 Stockholm", "Gothenburg", "Uppsala", "Copenhagen". The first alternative is clearly different from the others and is also the correct answer in this case.

- **Unanimously wrong**, meaning that neither of the provided 4 alternatives can be considered correct (the key).

90 (12.55%) MCQs were judged to be irrelevant with the distribution of reasons for irrelevancy depicted in Figure 7.

To summarize, the R-questions will be evaluated only on the MCQs that didn't have any problems so far. This includes $717 - 31 - 43 - 87 - 20 - 10 - 90 = 436$ MCQs (60.81%).

R1. Was only one alternative always correct, while the others were always wrong?
Answer: No, around 3 in 10 of the remaining MCQs (or 3 in 20 in total) had problems.

119 (16.6%) of the remaining 436 MCQs had more than one correct answer, which leaves us with 317 MCQs (44.21%) to be tested for the remaining conditions.

R2. Was the correct alternative always `a`?
Answer: No, a bit more than 3 in 10 of the remaining MCQs (or 3 in 20 in total) had `b`, `c`, or `d` as the correct alternative.

The distribution of positions of correct alternatives for the 317 MCQs remaining after R1 is provided in Figure 8. For **213 MCQs (29.71%)** the alternative `a` was correct, whereas all the other were wrong.

R3. Were the MCQs always ordered from the easiest to the hardest?
Answer: No, but for 27 texts they were!

For this part of the analysis, we have included all 317 MCQs with exactly one correct answer (no matter `a` or not) and without any problems spotted before R2. Notably, 6 texts have lost **all** their MCQs, so these 317 MCQs are spread over 90 out of the initial 96 texts.

We have then annotated each MCQ using the MCQ difficulty scheme outlined in Section 4.1 (and detailed in Appendices C.1, C.2, and C.3). The distribution of total MCQ difficulty is shown in Figure 9. Recall that the minimum possible MCQ difficulty is 3 points, whereas the maximum is 15 points. Each column in Figure 9 represents one of the 90 texts and each row is an MCQ generated for this texts. The MCQs are ordered in the order of generation from bottom to top (so the first row from the bottom indicates the first MCQ generated by GPT-3). Grey cells indicate MCQs excluded prior to R2.

If GPT-3 followed the prompt and ordered MCQs from easiest to hardest, one would expect

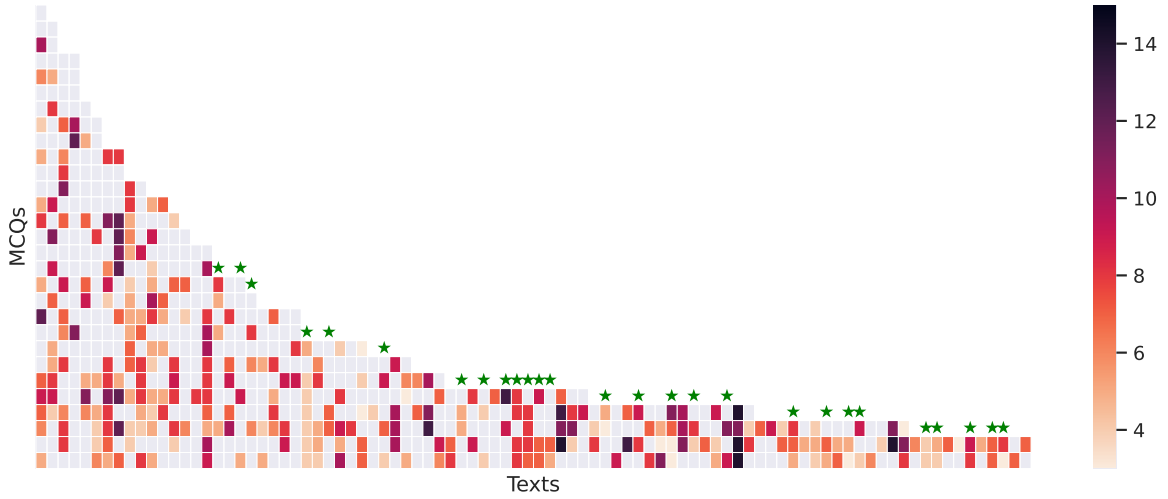


Figure 9: Heatmap of MCQ difficulty. Each column represents one of the 90 survived texts. Each cell in a column represents an MCQ generated by GPT-3 for this text. Grey cells indicate the MCQs excluded because of insufficient quality, whereas cells of other colors represent the accepted MCQs. Difficulty ranges from 3 to 15 and is represented by colors according to the legend on the right. Texts are ordered by their length (from the longest to shortest), which is directly proportional to the requested number of MCQs. MCQs are ordered by their generation order (those generated first reside in the first row from the bottom). The columns with green stars indicate the texts which had more than 1 accepted MCQ generated by GPT-3 in the order of their difficulty (from easiest to hardest).

the whole heatmap (except the grey cells) to follow the same coloring as the legend. The easier MCQs with difficulties close to the theoretically minimal 3 points should be at the bottom of the chart in light colors. The hardest MCQs should be on top of every column in dark colors (with difficulties close to theoretically maximal 15 points). However, Figure 9 shows neither this pattern, nor any pattern at all. Nevertheless, if we consider texts which have more than one survived MCQ **(72 out 90 texts)**, then MCQs were ordered in the non-decreasing order of difficulty for **27 texts** (marked with green stars in Figure 9).

6 Discussion and conclusions

GPT-3 has been able to generate around 30% of MCQs that conformed to all criteria (excluding ordering by difficulty), and 44% of MCQs which were of sufficient quality (also excluding the requirement that `a` is the correct answer). Together with our additional annotations, detailed with examples in Appendix E, these 44% of MCQs constitute *Quasi*, the first synthetic dataset of MCQs for testing reading comprehension of adult language learners of Swedish (available at `https://github.com/dkalpakchi/Quasi/blob/main/annotated/quasi.json`).

The fact that 44% of MCQs turned out to be of sufficient quality is impressive, given that (a) it is zero-shot, and (b) only 0.11% of GPT-3's training data was in Swedish. Although GPT-3 did not manage to order MCQs from easiest to hardest for most of the texts, the model could still generate MCQs of varying difficulty levels. The easiest MCQ scored theoretically minimal 3 points, whereas the hardest scored 14 (just 1 off from the theoretical maximum!).

That said, as 56% of MCQs turned out to be of insufficient quality (available at `https://github.com/dkalpakchi/Quasi/blob/main/annotated/poor_quality.json`), sometimes for subtle reasons, manual curation is not only desired, but is in fact *required* (not least to identify the correct alternative).

Why not ask GPT-3 to choose the correct alternative? One counter-argument is that it would consume more tokens, which leaves less tokens for MCQs, and leads to higher costs. Another reason is that there is no convincing argument why GPT-3 would be able to always provide the correct answer. If it could, then it should have been able to put it as alternative `a` all the time, which it did not. Furthermore, it should have been able to *always* generate only one correct answer, which it did not

do for 16.6% of MCQs. In fact, this finding is in line with the previously published evaluation of a BERT-based model for generating distractors in Swedish (Kalpakchi and Boye, 2021), where the most frequent reason for rejecting distractors was that they were not wrong (leading to more than 1 correct answer).

Could GPT-3 handle OCR errors, if there were any? Yes, it could! To give an example, one of the e-mail addresses in one of the texts was incorrectly recognized by the OCR system as *"ifhs.info Qimh.se"*, which we unfortunately didn't notice. GPT-3 was still able to generate *"ifhs.info@imh.se"* as one of the alternatives. This is most probably, because there was *"e-post:"* (eng. *"e-mail:"*) before this string, which the GPT-3's attention mechanism was able to capture. That said we didn't do any rigorous evaluation to quantify how well GPT-3 can mitigate OCR errors, so the caution is advised when trying to generalize from this insight.

Acknowledgments

This work was supported by Digital Futures within the project SWE-QUEST. We would like to thank Mariia Zyrianova for helpful discussions on the intricacies of the Kirsch scheme. We would also like to thank the anonymous reviewers for their helpful comments and suggestions.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Luke Gessler, Siyao Peng, Yang Liu, Yilun Zhu, Shabnam Behzad, and Amir Zeldes. 2020. AMALGUM – a free, balanced, multilayer English web corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5267–5275, Marseille, France. European Language Resources Association.

Xuanli He, Islam Nassar, Jamie Kiros, Gholamreza Haffari, and Mohammad Norouzi. 2022. Generate, annotate, and learn: NLP with synthetic text. *Transactions of the Association for Computational Linguistics*, 10:826–842.

Joan Jamieson, Stan Jones, Irwin Kirsch, Peter Mosenthal, and Carol Taylor. 2000. Toefl 2000 framework. *Princeton, NJ: Educational Testing Service*.

Dmytro Kalpakchi and Johan Boye. 2021. BERT-based distractor generation for Swedish reading comprehension questions using a small-scale dataset. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 387–403, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Dmytro Kalpakchi and Johan Boye. 2022. Textinator: an internationalized tool for annotation and human evaluation in natural language processing and generation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 856–866, Marseille, France. European Language Resources Association.

Irwin S Kirsch and Peter B Mosenthal. 1995. Interpreting the iea reading literacy scales. *Methodological issues in comparative educational studies: The case of the IEA reading literacy study*, pages 135–192.

Guillaume Le Berre, Christophe Cerisara, Philippe Langlais, and Guy Lapalme. 2022. Unsupervised multiple-choice question generation for out-of-domain q&a fine-tuning. In *60th Annual Meeting of the Association for Computational Linguistics*.

OECD. 2019. *PISA 2018 assessment and analytical framework*. OECD publishing.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

A Heuristic for choosing the number of generated MCQs

First we have calculated the average length of the sentence in characters for different sources of text (forum, news, blogs, etc), which could be extracted by multiplying $\bar{W}$ by $\bar{C}$ from Table 1. Then for each text T (belonging to category cat_T) we have calculated N_q^T as follows:

$$N_q^T = \alpha \frac{C_T}{\sum_{c \in cat_T} \bar{W}_c \cdot \bar{C}_c}, \quad (1)$$

where C_T is the number of characters in T, $\bar{W}_c$ ($\bar{C}_c$) is the average number of words (characters) per sentence in the corpus c belonging to the category cat_T (if a text did not belong to any category, we set cat_T to all categories), α is the assumed number of MCQs to be asked per sentence.

Choosing α is a bit tricky. In reality one can ask way more than 1 MCQ per sentence, but then not all sentences are worth asking even 1 MCQ. In hopes that these two groups cancel each other out, we have assumed $\alpha = 1$, meaning 1 MCQ per sentence, for the purpose of this article.

B Hyperparameter settings for GPT-3

We have used OpenAI's *text-davinci-003* model with the following generation hyper-parameters:

- temperature of 0.7
- "top p" (for nucleus sampling) of 1
- frequency and presence penalties of 0
- "best of" being equal to 1
- no custom stop sequences
- maximum length of 2048

C Scoring rules for process variables included in MCQ difficulty calculation

Recall that the core predictors found by Kirsch and Mosenthal (1995) are Type of Information (TOI), Type of Match (TOM), and Plausibility of Distractors (POD). The way these predictors were proposed to be operationalized is different depending on the nature of the provided textual material. More specifically, Kirsch and Mosenthal (1995) distinguished between the following two types of materials:

Corpus c	**Source**	$\lvert W_c \rvert$	$\bar{W}_c$	$\bar{C}_c$
Familjeliv	forum	885M	12.56	4.51
Flashback	forum	711M	12.92	4.67
Bloggmix	blogs	375M	13.82	4.69
Webbnyheter	news	87M	15.31	5.42
SVT	news	179M	13.65	5.40
Wiki	info	314M	10.67	5.63

Table 1: Corpus statistics for deciding the value of N_q. $|W|$ denotes a number of words in a corpus, $\bar{W}$ – the average number of words per sentence, $\bar{C}$ – the average number of characters per word

- **prose**, which refers to texts that consist of sentences grouped into paragraphs, in particular narrative and expository texts were considered;
- **documents**, meaning any kind of material where the structure of the document provides extra information for understanding the contents, for instance, e-mails (with address headers and footers), articles (with headlines), reports (with tables and graphs), advertisements, schedules, etc.

In the following section we discuss how we adapted the scheme proposed by Kirsch and Mosenthal (1995) to the needs of this article.

C.1 Type of Information

Kirsch and Mosenthal (1995) defined *Type of Information (TOI)* as the nature of what the readers are asked to identify in each given stem. The basic rule is that the more concrete the TOI, the easier the MCQ is, whereas more abstract TOI lead to more complex MCQs. The scoring rules are as follows.

- person, animal, or thing, score 1;
- amount, time, attribute, action, location, score 2;
- manner, goal, purpose, condition, or predicate adjective, score 3.
- cause, result, reason, evidence, or theme, score 4.
- equivalent, score 5.

The same rules apply for both prose and document materials.

C.2 Type of Match

Kirsch and Mosenthal (1995) defined *Type of Match (TOM)* in terms of processes used to relate information in the stem and the key to information in the textual material (prose or document). This is the most complex part of the scheme for evaluating MCQ difficulty, which we have simplified and took only the core aspect of it.

For the sake of brevity, we denoted the relation between a stem and the textual material as `S-T`, and the relation between a key and the textual material as `K-T`. For prose tasks, the majority of which were multiple-choice questions (MCQs),

Prose	Document	Score
when S−T **and** K−T are both LIT or SYN		1
when S−T **or** K−T requires LLTI, while the other requires LIT or SYN match		2
when S−T **and** K−T both require LLTI		3
when either S−T **or** K−T requires HLTI		4
when S−T **and/or** K−T requires HLTI, but the nature of the corresponding relation(s) needs to be defined by the reader	when S−T **and/or** K−T requires inferring a causal pattern or trend, or making a unique inference based on prior knowledge	5

Table 2: Scoring rules for Type of Match detailed for prose and document materials. S−T denotes the relationship between the stem and the text, and K−T between the key and the text, LIT stands for "literal", SYN – for "synonymous", LLTI – for "low-level text-based inference", whereas HLTI – for "high-level text-based inference".

we have adopted the scoring rules of Kirsch and Mosenthal (1995) as they are (see Table 2).

The scoring rules for document tasks were defined using many special-case rules. However, Kirsch and Mosenthal (1995) note that *many* of the document tasks did not use a multiple choice format, whereas in our case *all* tasks are guaranteed to be MCQs. Henceforth, instead of adopting, we chose to *adapt* by keeping as many applicable aspects of the rules for prose texts, as possible.

A clear similarity between prose and document scoring rules is that tasks requiring literal or synonymous match are still easier than those that need low-level text-based inference, which, in turn, are easier than those requiring a high-level text-based inference. Hence we decided *to keep the first 4 scoring rules as they are*.

One clearly different thing is the definition of the final level (when the MCQ should be awarded 5 points). We adopt this difference, but slightly adapt it, as shown in Table 2.

Unfortunately, Kirsch and Mosenthal (1995) do not provide clear definitions of what low-level or high-level inference mean, or where the border between synonymous match and low-level inference is. Hence, for the sake of this article, we have devised the following definitions based on examples of scoring MCQs, provided by Kirsch and Mosenthal (1995), and common sense.

Literal match (LIT) entails that the required information exists in the given textual material word-by-word. When applying this definition, it is allowed to ignore:

- question words/phrases, for instance, *"Vad"* (eng. *"What"*), *"Hur mycket"* (eng. *"How much"*), *"I vilket land"* (eng. *"In what country"*);
- articles, for instance *"en / ett"*, *"den / det / de"*, *"denna / detta / dessa"*;
- changes in the word form, when the word stem[6] remains the same (see examples in Table 3);
- changes between parts of speech (e.g., nominalization, adjectivization) when the word stem remains the same (e.g., *"drömmar"* and *"drömmer"*, *"samlas"* and *"samling"*)

For documents, there is a special case of LIT when matching information requires identifying structural part(s) in a document with a widely accepted structure. Such documents include, but are not limited to, e-mails, letters, blog posts, schedules. To exemplify, the question "Who is the sender of an e-mail?" requires the reader to locate the signature at the end of the email.

Synonymous match (SYN) encompasses cases when one word is substituted for another word with a similar meaning and the same grammatical features (part of speech, voice, inflection, number, etc). One or more such substitutions are allowed. Additionally, the following cases are included in this category, although they are not typically counted as synonyms in linguistics.

[6] not to be confused with the stem of an MCQ

TOM	Description	Example
LIT	conjugations of regular verbs	swe. *vänta / väntar / väntade / väntat* eng. *wait / waits / waited / waited*
LIT	conjugations of some irregular verbs to Present Simple	swe. *gå / går*, *stå / står*, *ge / ger* eng. *go / go*, *stand / stand*, *give / give*
LIT	noun inflections	swe. *bil / bilen / bilar / bilarna / bils / bilens / bilars / bilarnas* eng. *car / the car / cars / the cars / car's / the car's / cars' / the cars'*
LIT	adjective inflections	swe. *stor / stort / stora* eng. *large / large / large*
SYN	conjugations of strong verbs	swe. *finna / finner / fann / funnit*, *bryta / bryter / bröt / brutit* eng. *find / find / found / found*, *break / break / broke / broken*
SYN	most conjugations of irregular verbs	swe. *går / gick / gått*, *står / stod / stått*, *ger / gav / givit* eng. *go / went / gone*, *stand / stood / stood*, *give / gave / given*

Table 3: Examples of word form changes allowed different Type of Match levels

- Changes in the word form, when the word stem becomes different (see examples in Table 3).
- Using comparative and superlative adjectives, e.g., *"god / bättre / bäst"* (eng. *"good / better / best"*.
- When a word is matched to a part of a compound, e.g., *"kurserna"* in the stem and *"kursveckor"* in the text.
- Using abbreviations, e.g., *"tel."* for *"telefonnummer"*, *"kl."* for *"klockan"*, *"Feb"* for *"Februari"*.
- When numbers are written as words and vice versa, as well as colloquial names for numbers (*"a pair"* meaning 2).

Low-level text-based inference (LLTI) includes cases when:

- the required information needs to be "collected" from multiple sentences (e.g., coreference resolution);
- requires local (within sentence) reasoning (e.g., if the text says *"John is older than Mary"*, while one of the alternatives is *"Mary is younger than John"*);
- a word is substituted for another word with a different word stem, but a similar meaning, but different grammatical features (part of speech, voice, inflection, number, etc);
- a word is substituted by a phrase or vice versa;
- compounds (swe. *sammansättningar*) that are split into separate words, for instance, exchanging *"grundutbildning"* with *"grundläggande utbildning"* (note that these are very rare in English, but quite common in Swedish);
- hierarchical relationships, e.g. *"basketball"* and *"sport"*;
- the format needs to be recognized, e.g. that *"name@example.com"* is an e-mail;
- a non-matching word denotes whether the information should be included in/excluded from a document (or a part of a document).

High-level text-based inference (HLTI) when it is required to link multiple paragraphs of text. These cases include, but are not limited to:

- counting entities (if they are not already counted in the text), such as in the stem *"How many countries are represented in the event"*;
- reading between the lines to find out the information, as in *"Why did John write this letter?"*;
- using specific prior knowledge about content or structure of the text, for instance, when one writes *"Otto, 27"* at the end of the post on social media means that *"27"* is most probably his age, or when it's written *"Opening*

hours 11 - 21", it means that the closing time is 21:00;

- asking whether the information is included in the text, e.g. *"What is not a hobby of John?"*.

The last level, for the score of 5, requires the reader to define the nature of the S−T and/or K−T relations. In particular, this included the cases when the reader needs to

- provide an interpretation of a phrase based on the information in the text, e.g., *"Vem av personerna i texten är mest förespråkare för förbud?"* (eng. *"Which of the people in the text advocates the most for the ban?"*);

- recognize the stance of a person in the text, e.g. *"Hur resonerar Joel Marklund kring ekologiska produkter?"* (eng. *"How does Joel Marklund reason about the ecological products?"*).

For all TOM levels, pleonasms (meaning redundant linguistic expressions that are unnecessary to comprehend the stem) should be *ignored*. For instance, consider the key is *"make demands on the children but show them love"*. The pronoun *"them"* is only there because of grammar, otherwise it is apparent from the context that love should be directed towards children, even without the pronoun. Note that in MCQs expressions might become pleonastic with respect to the given alternatives. For instance, consider the sentence *"Welcome to the interview on Wednesday 6/2 at 15:00"* and the following MCQ.

> What is the date for the interview?
>
> a) Thursday 7/2
>
> b) Wednesday 6/2
>
> c) Wednesday 15/2
>
> d) Thursday 6/2

Obviously, the word *"date"* is not mentioned in the text and one needs to know what the date is, so this appears to be a case of HLTI. However, given the alternatives, one doesn't need to understand the word *"date"* and suddenly the match between the stem and the text gets downgraded to LIT.

To give another example consider the following e-mail.

> Hi,
>
> I was forced to pay $20 extra for the delivery of the laptop, which I think is unacceptable!
>
> Best regards, Martin Jones

If we analyze the MCQ below at a first glance, *"Martin Jones"* is not mentioned in the sentence about extra $20 payment. Instead *"I"* there should be resolved to *"Martin Jones"*, so it seems like a case of LLTI. However, Martin Jones is the only person that is in fact mentioned in the text, so mention of his name in the stem becomes pleonastic and hence the MCQ again gets downgraded to LIT.

> How much was Martin Jones forced to pay?
>
> a) $20
>
> b) $15
>
> c) $40
>
> d) $2

What these two examples show is that the judgement of Type of Match level if *extremely* text-dependent and one and the same MCQ could get different TOM-score, depending on the text at hand.

C.3 Plausibility of Distractors

Inspired by (Kirsch and Mosenthal, 1995), we make use of an implicit tree structure for both prose and document materials. Each node of such a tree should contain a unit of information that cannot be split further into independent units. The only type of nodes in prose texts are paragraphs, whereas nodes in documents are generalized paragraphs by nature, but could also contain more structured and/or graphical material (such as charts, tables, maps, lists, etc).

Given the surface form of the correct answer, we define *the answer node* (AN) as the first node in the BFS traversal of the tree corresponding to the textual material, containing information supporting the correct answer.

Since *many* of the document tasks did not use a multiple choice format, as noted by Kirsch and Mosenthal (1995), the rules for scoring POD for document materials must be adapted. If the format is not MCQs, then it is only relevant to look into *distracting information* in the text, i.e., pieces of

Prose	Document	Score
	There is no distracting information in the text	1
	DIS are LIT **or** SYN match to the information **not** in AN	2
	DIS represent PII **not** based on information related to AN	3
	One DIS contains information that is related to the information in AN	4
	Two or more DIS contain information that is related to the information in AN	5
	One or more DIS represent PII based on information outside of the text	5

Table 4: Scoring rules for Plausibility of Distractors detailed for prose and document materials. DIS stands for "distractor(s)", LIT – for "literal", SYN – for "synonymous", PII – for "plausible invitied inferences", and AN – for "answer node".

text that provide plausible grounds, although they are still not correct. In stark contrast, MCQs already provide a number of alternatives, which the reader is forced to choose between. Hence the distracting information is only relevant if one of the distractors in the alternatives relies on it. Keeping that in mind, we have adapted the POD scoring rules for prose texts to the document texts by generalizing from paragraphs to nodes (see Table 4).

D Grammatical error types

The following is a list of grammatical error types, which we adopted for this article. Note that this is *not* an exhaustive list of grammatical error types, but very much specific to the synthetic data at hand.

- wrong verb forms, such as *"meddelar"* in the stem *"Vilka problem kan man meddelar om man har ett akut problem?"*;
- wrong noun forms, such as wrong case;
- wrong prepositions, such as *"hos anläggningen"* instead of *"i anläggningen"*;
- wrong grammatical agreement (AGR), such as *"en krav"* instead of *"ett krav"*, or *"det minst antalet"* instead of *"det minsta antalet"*;
- syntax errors, most often errors in constructions of sentences, e.g., *"Är kursbok och arbetsmaterial ingår i kursavgiften?"* (eng. *"Are the course book and work material includes in the course fee?"*);
- spelling errors, such as *"addressedes"* instead of *"addresserades"*, or *"städt"* instead of *"städat"*;
- wrong lexical choice, when a word should not be used in the provided context, for instance the stem *"Vilka huvudroller är med i Lyckliga dagar?"* (eng. *"Which main roles participate in The happy days"*), or using the pronoune *"deras"* instead of *"sina"*;
- logical errors, when a word/phrase is used in a way that does not conform to its properties, for instance *"cykelbana"* in the stem *"I vilken sorts transportmedel finns en cykelbana?"* (eng. *"In what kind of transport does the bicycle lane exist?"*), or *"lokalen"* in the alternative *"lokalen tar för lång tid att spela"* (eng. *"the premises take too long to play"*);
- tautology, such as *"poetiska dikter"* (eng. *"poetic poems"*).

E Additional annotations for Quasi

In addition to the annotations necessary for evaluating the difficulty of each MCQ in Quasi, we also provide the following annotations (exemplified in Figure 10):

- the phrase/sentence that serves as the basis for the key;
- the phrase/sentence that serves as the basis for each distractor (for those distractors that actually use information from the text);
- the answer nodes that the learner must read in order to answer the question.

Hej,

Min mobil verkar äntligen fungera nu när jag har betalat fakturapåminnelsen med en förseningsavgift. Men att jag dessutom tvingades betala 160 kronor extra i öppningsavgift tycker jag är upprörande. Vill bara att ni ska få veta.

Hälsningar,
Eva-Lena Hansson

2. Varför skrev Eva-Lena Hansson detta brev?

a) För att berätta att hennes mobil äntligen fungerade

b) För att berätta att hon hade betalat för sent

c) För att berätta att hennes mobil inte fungerade

d) För att berätta att hon hade betalat för mycket

Figure 10: Example of additional annotations in Quasi. The bases for each alternative are highlighted in the text with the corresponding color. The required answer nodes are underlined. The key (correct alternative) is highlighted in bold.

Automatic Closed Captioning for Estonian Live Broadcasts

Tanel Alumäe
Tallinn University of Technology
tanel.alumae@taltech.ee

Joonas Kalda
Tallinn University of Technology
joonas.kalda@taltech.ee

Külliki Bode
Estonian Association of the Hard of Hearing
kylliki.bode@gmail.com

Martin Kaitsa
Estonian Public Broadcasting
martin.kaitsa@err.ee

Abstract

This paper describes a speech recognition based closed captioning system for Estonian language, primarily intended for the hard-of-hearing community. The system automatically identifies Estonian speech segments, converts speech to text using Kaldi-based TDNN-F models, and applies punctuation insertion and inverse text normalization. The word error rate of the system is 8.5% for television news programs and 13.4% for talk shows. The system is used by the Estonian Public Television for captioning live native language broadcasts and by the Estonian Parliament for captioning its live video feeds. Qualitative evaluation with the target audience showed that while the existence of closed captioning is crucial, the most important aspects that need to be improved are the ASR quality and better synchronization of the captions with the audio.

1 Introduction

Deaf and hard of hearing (DHH) individuals face significant barriers when it comes to accessing live television broadcasts. Without closed captioning, they are unable to fully understand and engage with the content being presented. An automatic closed captioning system for live TV broadcasts would help to address this issue and provide DHH individuals with greater access to the same information and entertainment as their hearing counterparts. Closed captioning is not only beneficial for DHH individuals, but also for those who may have difficulty hearing the audio on their television due to background noise or other factors.

Until the beginning of 2022, Estonian Public Television (ETV) provided DHH-focused subtitles for some pre-recorded native language programmes, but not for live programmes. From March 2022, captions generated using automatic speech recognition (ASR) technology were added to the majority of live native-language programmes, such as news and talk shows. The same technology is used to provide closed captions to the live streams of the Estonian parliament sessions (see Figure 1). This paper describes the system used to generate the subtitles. We provide information on the architecture of the system, its different components, their training data and performance. We also summarise the results of a qualitative evaluation of the live captioning system carried out with the target audience, and discuss how the system could be improved.

Figure 1: Closed-captioned live YouTube stream of the Estonian parliament.

The reported system is free and available under open-source license [1].

2 Previous Work

Real-time captioning systems based on speech recognition have been in use for several decades. Initially, such systems relied on so-called respeakers - trained professionals who repeat what they hear in the live broadcast in a clear and ar-

[1] https://github.com/alumae/kiirkirjutaja

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 492–499

ticulate manner (Evans, 2003; Imai et al., 2010; Pražák et al., 2012). This allows supervised speaker adaptation of ASR acoustic models to be used, resulting in very accurate output. In some use cases, re-speakers also simplify and rephrase the original speech, instantly check and correct the resulting captions, and insert punctuation symbols. In some captioning systems, ASR is applied directly to the speech in the live programme, but a human editor is used to correct the ASR errors (Levin et al., 2014). However, training re-speakers and real-time editors is a long and expensive process. In addition, several re-speakers and/or editors are usually required, as one person cannot usually work continuously for more than two hours without a break.

As the quality of ASR systems has improved rapidly in recent years, there are more and more cases where an ASR system is used directly to produce subtitles without any post-processing. For example, ASR-based captions in multiple languages are available in online meeting platforms such as Zoom, Skype and Teams. Moreover, YouTube offers captioning for live streams, albeit exclusively in English at the time of writing. Streaming ASR for Estonian is available through several commercial vendors; however, recent evaluations have demonstrated that the ASR quality provided by these services falls short compared to the models developed at Tallinn University of Technology (Vapper, 2023).

3 Closed Captioning System

3.1 Architecture

Our closed captioning system consists of the following components:

1. Speech input: speech is either read from standard input (as a 16-bit 16 kHz PCM audio stream) or from a URL. Any stream type supported by *ffmpeg* is allowed, including video streams.

2. Audio stream is segmented into 0.25 second chunks and processed by a voice activity detection model which detects speech start and endpoints in the stream. We use the open source Silero VAD model (Silero Team, 2021), available under the MIT License;

3. Speaker change detection model indicates likely speaker change points in speech segments (see Section 3.2);

4. Each speaker turn is processed by a language identification module that filters out segments that are likely not in Estonian (Section 3.3);

5. Speech recognition, resulting in a stream of words tokens (Section 3.4);

6. Inverse text normalization (mostly converting text to numbers), implemented using hand-written finite state transducer rules using the Pynini library (Gorman, 2016);

7. Insertion of punctuation symbols and subsequent upperacasing (Section 3.5);

8. Confidence filter that hides decoded words that are likely to be incorrect (Section 3.6);

9. Presentation: displaying the captions or sending them to the API endpoint selected by the user (Section 3.7).

3.2 Speaker Change Detection

In order to make captions for dialogue more legible speaker change points need to be marked by a symbol such as "-". To detect change points we use an online speaker change detection model[2] which treats this as a sequence classification problem and labels each frame with either 1 or 0 depending on whether a speaker change happened or not.

The model is trained on an Estonian broadcast dataset detailed further in Section 3.4.1. Training is done on samples from speech segments with random lengths between 10 and 30 seconds. Background noise and reverberation are added to each segment both with a probability of 0.3. Background noises come from the MUSAN corpus (Synder et al., 2015). For reverberation, we used small and medium room impulse responses as well as real room impulse responses (Ko et al., 2017; Szöke et al., 2019). A classification threshold is learned on a 1-hour development split.

The model uses 1280-dimensional features obtained from a Resnet-based extractor (Alumäe, 2020) which is pre-trained on VoxCeleb2 (Chung et al., 2018). This is followed by two long short-term memory (LSTM) layers both with 256-dimensional hidden layers. A 1-second label delay is used since the model needs to see past the current frame to predict a change point. We use

[2] https://github.com/alumae/online_speaker_change_detector

a collar-based loss function that encourages the model to predict a single positive frame in a 250ms neighborhood of an annotated change point. This training method has been shown to outperform the standard binary cross-entropy loss for the SCD task (Kalda and Alumäe, 2022). A further benefit of this loss function is that the model outputs develop peaks concentrated in a single frame. This removes the need for post-processing to find the exact timestamps of change points and decreases overall latency.

3.3 Language Identification

Broadcast news programs often contain foreign language segments, such as studio or field interviews. For those segments, no captions should be shown, since an Estonian ASR system doesn't produce meaningful output for speech in other languages. Furthermore, foreign language video segments in television news programs often already have Estonian subtitles and automatic captions would interfere with them.

For filtering out non-Estonian speech segments, we first process the first three seconds of every speech turn using the open source Silero language identificaton model (Silero Team, 2021), available under the MIT License. During the initial development phase, we found that the first 3 seconds are sometimes unreliable for language detection, since they often contain hesitation and/or other paralingusitic speech sounds that confuse the language detection model. Therefore, if a turn is rejected based on the first three seconds, another test is performed using the first five seconds of the turn. If this test also indicates that the speech is not in Estonian, the whole speech turn is ignored by the rest of the pipeline and no captions are produced for this speaker turn. Of course, this assumes that a speaker doesn't change the language during a single turn which might not always be true.

The language classifier that we use discriminates between 95 languages and claims 85% validation accuracy. However, we are not interested in the actual language spoken in the segments, but only in the the fact whether the segment is in Estonian or not. This allows us to use a simple method to increase the robustness of the language classifier. Namely, we assume that our system is always used on streams where the input language is mostly in Estonian, which means that the prior probability of Estonian is much higher than the default uniform probability $P_u(l) = 1/95$. Therefore, we "fix" the conditional probability distribution $P(l|x)$ returned by the language identification model for input segment x to use the appropriate prior:

$$P'(l|x) = \frac{\frac{P'(l)}{P_u(l)} \times P(l|x)}{Z}$$

where Z is a normalizing factor and $P'(l)$ is the prior probability for languages:

$$P'(l) = \begin{cases} P'(l = \text{et}), & \text{if } l = \text{Estonian} \\ (1 - P'(l = \text{et}))/94, & \text{otherwise} \end{cases}$$

Based on small-scale finetuning, we use a prior probability $P'(l = \text{et}) = 0.5$ for Estonian.

Source	Amount (h)
Broadcast speech	591
Spontaneous speech (Lippus, 2011)	53
Elderly speech corpus (Meister and Meister, 2022)	49
Talks, lectures	38
Parliament speeches	31
Total	761

Table 1: Acoustic model training data.

Source	Tokens (M)
ENC19 Web Scrape	526
ENC19 Ref. Corpus	185
ENC19 Wikipedia	35
OpenSubtitles	98
Speech transcripts	6.1
Subtitles from ETV	3.8
Total	854

Table 2: Language model training data.

3.4 Speech Recognition

3.4.1 Data

Speech data that is used for training the speech recognition acoustic model is summarized in Table 1. Only the duration of the segments containing transcribed speech is shown, i.e., segments containing music, long periods of silence and untranscribed data are excluded.

Most of the training data has been transcribed by our lab in the last 15 years (Meister et al., 2012), except the Corpus of Estonian Phonetic Corpus of Spontaneous Speech that originates from the University of Tartu (Lippus, 2011).

Textual data used for training the language model (LM) is listed in Table 2. Most of the data originates from the subcorpora of the Estonian National Corpus 2019 (ENC2019) (Kallas and Koppel, 2019): Estonian web, a reference corpus containing balanced data from the web, newspapers and books, and Estonian Wikipedia. We also use all available Estonian data from the OpenSubtitles corpus (Lison and Tiedemann, 2016) and scraped DHH subtitles from ETV.

Before using the text data for LM training, text normalization is performed. Texts are tokenized, split into sentences and recapitalized, i.e., converted to a form where names and abbreviations are correctly capitalized while normal words at the beginning of sentences are written in lower case. This is done with the help of the EstNLTK morphological analyzer (Laur et al., 2020). Numbers and other non-standard words are expanded into words using hand-written rules.

3.4.2 Models

The ASR model is implemented using Kaldi (Povey et al., 2011). The acoustic model is a factored time-delay neural network (TDNN-F) acoustic model (Povey et al., 2018) with six convolutional layers and 11 TDNN-F layers. The acoustic model has around 17 million parameters. Online speaker adaptation is done using i-vectors. We use standard Kaldi multi-condition data augmentation (Ko et al., 2017) for acoustic training data: training data is 3-fold speed perturbed, and the speed perturbed data is in turn augmented with reverberation, various environment sounds, music or babble noise from the MUSAN corpus (Synder et al., 2015). This increases the amount of training data by 15-fold in total. The acoustic model is trained for four epochs on the augmented data.

The LM of the system uses 200 000 compound-split units (i.e., compound words are broken to constituents). It is an interpolation of 4-gram submodels trained on each of the subcorpora, with interpolation coefficients optimized on development data. The final model is pruned so that the resulting HCLG transducer would allow decoding with 16 GB of RAM. After decoding, we apply out-of-vocabulary (OOV) word recovery to reconstruct the orthographic transcripts of the decoded unknown words. Compound words are reconstructed from the decoded constituents using a hidden-even n-gram model (Alumäe, 2007). Various specifics of language modeling are described in more details in (Alumäe et al., 2018).

	WER
TV news	8.5
Talkshows	13.4
Press conferences	8.1

Table 3: Word error rate of the ASR system on various speech data.

We validated the performance of the models on a dedicated test set collected especially for this project. It consists of TV main evening news, casual TV talkshows, and press conferences of the Tallinn city council and the state's health board, with a total duration of 12 hours. Table 3.4.2 shows the word error rate (WER) of the ASR system on each subcorpus. As can be seen, TV news and press conferences produce noticably less ASR errors than talkshows, which is probably related to the higher degree of spontaneousness in talkshow speech.

The decoding module is implemented using a forked version of the Vosk Speech Recognition Toolkit[3] that supports word timestamps for intermediate recognition hypotheses.

Closed captions on television generally do not offer verbatim speech transcriptions, particularly for spontaneous speech. Elements such as repetitions, hesitations, pause fillers, false starts, and interjections are typically omitted from the captions, and sentences are reformatted to ensure grammatical correctness. Presently, our system lacks any modules to implement such modifications on the generated ASR transcripts. Only filled pauses and hesitations are excluded from the captions, since they are not transcribed in the ASR training data.

3.5 Punctuation Insertion

In order to make the captions more readable, the decoded stream of words is enriched with punctuation symbols. This is done using an LSTM model[4]. The model is trained on a mixture of speech transcripts from our ASR training corpus and a random sample of the LM training data, totalling in around 50 million words. The punctuation model operates on BPE-tokenized text, using a BPE vocabulary of 100K tokens. The model first projects the input tokens into 512-dimensional embeddings and then applies four unidirectional

[3] https://github.com/alphacep/vosk-api

[4] https://github.com/alumae/streaming-punctuator

LSTM layers, with a hidden layer dimensionality of 512. For token corresponding to word endings, the most likely punctuation symbol is predicted from the vocabulary of [*None*, ".", ",", "?", "!"]. A label delay of two is used, i.e., at each time step, the model predicts a punctuation symbol for a token two timesteps in the past. This effectively allows the model to predict a punctuation symbol, given the past tokens and two upcoming tokens.

The model was validated on the transcripts of the ASR validation set and resulted in a F1 score of 72%, micro-averaged across all punctuation marks.

3.6 Confidence Filter

In some situations, such as severe background noise, overlapped speech or very spontaneous speech, the quality of the ASR output degrades significantly. In such cases, it is preferable not to show any captions at all, since they are practically useless for understanding the content of the speech and bring a lot of confusion to the viewer. Therefore, the closed captioning system includes an additional component that tries to hide captions segments that are likely wrong.

The ASR decoder that we use outputs word confidence values for all decoded tokens. The confidence scores are computed by the Kaldi decoder from the confusion network of the Minimum Bayes Risk (MBR) decoding result (Xu et al., 2011). Since such confusion scores are often not very reliable, the captioning system observes the averaged confidence scores of the words calculated over a five word window, and hides words whose averaged confidence score falls below a threshold (we use a threshold of 0.75). Evaluating, finetuning and calibration of this component remains currently for future work.

3.7 Presentation

The system can present the generated captions in a variety of formats and modes. Currently, it supports several commercial captioning delivery platforms as well as YouTube live streaming. Most media streaming platforms that support closed captioning expect word-by-word captions: i.e., captions should be provided on a word-by-word basis (possibly with a timestamp), and words already displayed cannot be changed. This poses some challenges for our captioning system, as several factors cause the final part of the caption to change dynamically: new words coming from the decoder may cause already decoded words to change (e.g. due to word to number conversion), punctuation may be inserted before the already decoded word (due to the two-word label delay of the punctuation model). For this reason, the caption presentation module includes functionality to delay the final output of generated words to the currently used subtitle transmission platform until it is certain that the word won't change. This (and the delay caused by the speaker change detection model and the language identification model) results in a delay of approximately 3-5 seconds relative to the speaking time of the words, which can be mitigated by also delaying the transmission of the multimedia stream. For those presentation modes that allow dynamically changing captions, a much lower delay or approximately 2 seconds is possible.

Figure 2: Closed-captioned ETV talk show.

4 Integrations

At the time of writing this paper, the closed captioning system is used by the Estonian Public Television (ETV) and by the Estonian parliament.

In ETV, the captioning system runs continuously, but the captions are actually delivered only to specific native-language programs (see Figure 2). The system outputs captions on a word-by-word basis to special caption transmission software that formats the words into caption lines and blocks. Due to the approximately 5-second delay in the video signal caused by the encoding process, the captions and video are roughly synchronized, but the synchronization is currently not exact: captions tend to be delayed at the beginning of a speaker's turn and arrive relatively faster at the end of a turn.

Closed captions are transmitted on a dedicated DHH digital closed captioning channel and are not displayed by default. End users can enable closed

captioning from the user interface of their device.

5 Qualitative Evaluation

5.1 Introduction

In order to better understand, how the automatically generated captions on ETV are used and experienced, and what are the most outstanding shortcomings, we conducted a qualitative evaluation with the intended focus group of the technology. The purpose of this study was to investigate the following research questions:

- How often do DHH individuals use the ASR-generated closed captions on ETV?
- How do the closed captions improve the quality of life of DHH individuals?
- Which aspects of the system need to be improved?

5.2 Methodology

This research study used semi-structured interviews to gather information. Since most of the study participants were DHH individuals, three of the interviews were conducted via e-mail, one via a chat application, and only one via telephone. Additional user feedback was collected via Facebook in response to a call for comments by a person followed by a large DHH community.

Participants were sought through personal contacts of the researcher. Four of the participants were hard-of-hearing individuals and one didn't have any significant hearing loss. Data collection took place in January 2023.

5.3 Findings

Three participants with hearing impairments acknowledged that their hearing loss prevents them from understanding speech on television, even when using a hearing aid. One individual with hearing impairment mentioned that she could comprehend the speech if the TV volume were significantly increased, but she refrains from doing so as she is the only person with hearing impairment in her family. All the DHH participants reported that they activate automatic subtitles whenever they watch live ETV broadcasts daily. The only participant without hearing loss revealed that he watches programs with automatic subtitles multiple times a week when ambient household noise or background conversations make it challenging to hear the television audio.

All the participants highlighted the importance of having subtitles. Several people reported that it enabled them to watch television with their family. One participant expressed that subtitles helped her feel included in society and enabled her to stay better informed about events occurring in the country.

All participants stressed that the most crucial aspect of the existing captioning system requiring improvement is its accuracy. One participant highlighted that the quality of captions is currently good for TV presenters, but often falls short for "ordinary people" (i.e. interviewees on news and talk shows). One person explained that although it is sometimes difficult to understand what is actually being said (due to ASR errors), it is still important to have the captions. The second aspect that was often highlighted was that in the current captioning system, the captions are often not well synchronized with the audio (as opposed to manually created subtitles). Other issues raised include misrecognition of named entities, poor marking of speaker turns, occasional dropping of the captions (i.e., when the confidence filter is activated) and the fact that subtitles sometimes interfere with other information on the screen, such as speaker names. Several respondents also noted that subtitles are not currently available for all native language broadcasts.

6 Conclusion and Future Work

This paper described an ASR-based realtime closed captioning system for Estonian broadcasts. The system consists of several open-source components and is currently used for providing captions to Estonian public television native language broadcasts and for captioning the live streams of the Estonian parliament.

Qualitative evaluation with the hard-of-hearing focus group showed that providing captions to live TV broadcasts is of high importance to this community. The study emphasized that it is urgent to further improve the ASR quality of the closed captions and to improve the synchronization between audio and captions.

We are currently working on several aspects of the system that would address some problems highlighted in the qualitative evaluation. First, we are preparing to migrate to end-to-end streaming transducer ASR models that would provide improved accuracy with relatively low latency. We

are also experimenting with integrating the decoding of punctuation symbols to the main ASR model, since currently the separate punctuation symbol insertion model is a source of around two second latency in subtitle presentation. Also, we have already implemented modifications to the system that would allow exact synchronization between the audio and displayed captions.

Acknowledgments

This research has been supported by the Centre of Excellence in Estonian Studies (CEES, European Regional Development Fund). The authors acknowledge the TalTech supercomputing resources made available for conducting the research reported in this paper.

References

Tanel Alumäe. 2007. Automatic compound word reconstruction for speech recognition of compounding languages. In *NODALIDA 2007*.

Tanel Alumäe. 2020. The TalTech system for the VoxCeleb Speaker Recognition Challenge 2020. Technical report. https://www.robots.ox.ac.uk/~vgg/data/voxceleb/data_workshop_2020/taltech.pdf.

Tanel Alumäe, Ottokar Tilk, and Asadullah. 2018. Advanced rich transcription system for Estonian speech. In *Baltic HLT 2018*, pages 1–8.

J. S. Chung, A. Nagrani, and A. Zisserman. 2018. VoxCeleb2: Deep speaker recognition. In *Interspeech 2018*.

MJ Evans. 2003. Speech recognition in assisted and live subtitling for television. *R&D White Paper WHP*, 65.

Kyle Gorman. 2016. Pynini: A Python library for weighted finite-state grammar compilation. In *SIGFSM Workshop on Statistical NLP and Weighted Automata 2016*, pages 75–80.

Toru Imai, Shinichi Homma, Akio Kobayashi, Takahiro Oku, and Shoei Sato. 2010. Speech recognition with a seamlessly updated language model for real-time closed-captioning. In *Interspeech 2010*.

Joonas Kalda and Tanel Alumäe. 2022. Collar-aware training for streaming speaker change detection in broadcast speech. In *Odyssey 2022: The Speaker and Language Recognition Workshop*.

Jelena Kallas and Kristina Koppel. 2019. Estonian National Corpus 2019. https://doi.org/10.15155/3-00-0000-0000-0000-08565L.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur. 2017. A study on data augmentation of reverberant speech for robust speech recognition. In *ICASSP 2017*.

Sven Laur, Siim Orasmaa, Dage Särg, and Paul Tammo. 2020. EstNLTK 1.6: Remastered Estonian NLP pipeline. In *LREC 2020*, pages 7152–7160.

Kirill Levin, Irina Ponomareva, Anna Bulusheva, G Chernykh, Ivan Medennikov, Nickolay Merkin, Alexey Prudnikov, and Natalia Tomashenko. 2014. Automated closed captioning for Russian live broadcasting. In *Interspeech 2014*.

Pärtel Lippus. 2011. *The acoustic features and perception of the Estonian quantity system*. Ph.d. thesis, University of Tartu.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *LREC 2016*.

Einar Meister and Lya Meister. 2022. Estonian elderly speech corpus–design, collection and preliminary acoustic analysis. *Baltic Journal of Modern Computing*, 10(3):360–371.

Einar Meister, Lya Meister, and Rainer Metsvahi. 2012. New speech corpora at IoC. In *XXVII Fonetiikan päivät 2012*.

Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohamadi, and Sanjeev Khudanpur. 2018. Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Interspeech 2018*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Vesely. 2011. The Kaldi speech recognition toolkit. In *ASRU 2011*.

Aleš Pražák, Zdeněk Loose, Jan Trmal, Josef V Psutka, and Josef Psutka. 2012. Captioning of live TV programs through speech recognition and re-speaking. In *International Conference on Text, Speech and Dialogue*, pages 513–519. Springer.

Silero Team. 2021. Silero VAD: pre-trained enterprise-grade voice activity detector (VAD), number detector and language classifier. https://github.com/snakers4/silero-vad.

David Synder, Guoguo Chen, and Daniel Povey. 2015. MUSAN: A music, speech, and noise corpus. ArXiv:1510.08484.

I. Szöke, M. Skácel, L. Mošner, J. Paliesek, and J. Černocký. 2019. Building and evaluation of a real room impulse response dataset. *IEEE Journal of Selected Topics in Signal Processing*, 13(4):863–876.

Silver Vapper. 2023. Benchmarking Estonian ASR solutions. Speech Technology Workshop: https://haldus.taltech.ee/sites/default/files/2023-03/Benchmarking_Estonian_ASR_solutions.pdf.

Haihua Xu, Daniel Povey, Lidia Mangu, and Jie Zhu. 2011. Minimum Bayes risk decoding and system combination based on a recursion for edit distance. *Computer Speech & Language*, 25(4):802–828.

The Effect of Data Encoding on Relation Triplet Identification

Steinunn Rut Friðriksdóttir
Department of Computer Science
University of Iceland
Iceland
srf2@hi.is

Hafsteinn Einarsson
Department of Computer Science
University of Iceland
Iceland
hafsteinne@hi.is

Abstract

This paper presents a novel method for creating relation extraction data for low-resource languages. Relation extraction (RE) is a task in natural language processing that involves identifying and extracting meaningful relationships between entities in text. Despite the increasing need to extract relationships from unstructured text, the limited availability of annotated data in low-resource languages presents a significant challenge to the development of high-quality relation extraction models. Our method leverages existing methods for high-resource languages to create training data for low-resource languages. The proposed method is simple, efficient and has the potential to significantly improve the performance of relation extraction models for low-resource languages, making it a promising avenue for future research.

1 Introduction

Relation extraction (RE) is a task in the field of natural language processing, aimed at identifying and extracting semantically meaningful relationships between entities present in text. A relation is generally extracted as an ordered triple (E_1, R, E_2) where E_1 and E_2 refer to the entity identifiers and R refers to the relation type. This task holds great significance in several practical applications, including information retrieval, knowledge management, and question-answering systems, among others (for a recent review, see (Yan et al., 2021)). The increasing availability of unstructured text data on the web has only served to underline the importance of relation extraction, as there is a pressing need to convert this data into structured information that can be easily accessed and analyzed. This challenge is especially pertinent for low-resource languages like Icelandic, where the limited availability of annotated data presents a significant impediment to the development of high-quality relation extraction models.

Relation extraction methods for English have evolved over the years, with early methods relying on hand-crafted rules, patterns, and statistical analyses (Soderland et al., 1995; Carlson et al., 2010; Kambhatla, 2004; Jiang and Zhai, 2007). With the advent of deep learning and the availability of large annotated corpora, more sophisticated methods have emerged (Liu et al., 2013; Xu et al., 2016; dos Santos et al., 2015). Deep learning models have shown promising results in extracting relationships between entities, outperforming traditional methods. Current state-of-the-art model, REBEL (Cabot and Navigli, 2021), performs joint relation extraction[1]. The progress in relation extraction for English has demonstrated the potential for using advanced techniques to extract meaningful relationships from large amounts of text data.

The challenge of developing effective RE methods for languages like Icelandic lies in the scarcity of annotated data. The performance of machine learning models heavily relies on the availability of large amounts of annotated training data. The limited availability of annotated text in low-resource languages creates a major challenge for training high-quality relation extraction models. To overcome this challenge, one way is to study methods to efficiently create training data based on existing methods for English. Our main question is whether models from high-resource languages can be used to efficiently create training and testing data for low-resource languages.

In this paper, we present a novel method for efficiently creating relation extraction data for low-

[1]That is, entity extraction and relation extraction are not two separate processes.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 500–507

resource languages like Icelandic. Our method is based on replacing entities in the text with unique identifiers, translating the text to a high-resource language using machine translation, and then replacing the entities back in the translated text. Finally, we perform relation extraction on the translated text to obtain the relationships between entities. Our method is simple and only requires the location of entities in the text and a machine translation model. This approach leverages the availability of specialized models in high-resource languages to create training data for low-resource languages, thereby addressing the challenge posed by the scarcity of annotated data. Our method has the potential to significantly improve the performance of relation extraction models for low-resource languages, making it a promising avenue for future research.

2 Purpose

In a previous paper, we proposed a way of bootstrapping RE training data for low-resource languages (LRL) using a combination of machine translation and open RE methods (Friðriksdóttir et al., 2022). By automatically translating the LRL data into English, we were able to feed it directly into the high-resource language SOTA model before translating the relation triplets back to the LRL where it can serve as training data for a new LRL model.

While this method showed potential, the resulting data was jumbled by errors in translation. Some examples of this include people's names being perturbed by the multilingual translation model (resulting in *Alfreð* being changed into *Alfredo*, *Sveinn* into *Sweene* etc.), entities getting directly translated and thereby loosing their meaning (such as when the Icelandic name *Erlendur* gets directly translated as *foreign*) and unfortunate translation mishaps (such as when *Dauðarósir*, a novel by the Icelandic author Arnaldur Indriðason, gets translated as *Deathly Hallows*, a real novel by a different author).

In this paper, we hypothesize that these translation errors can be ameliorated by encoding the entities within the data before it gets translated, and then decoding them before they get sent into the high-resource RE model. Whereas the proposed method remains the same, this extra step in preprocessing the input data should result in more accurate predictions made by the RE model, which in turn makes for better training data.

3 Previous Work

Machine-translation has previously been used to create cross-lingual named entity recognition (NER) datasets, which led to improvement in NER for several languages (Dandapat and Way, 2016; Jain et al., 2019). In these earlier works, the text was translated directly, without any modifications, and the entities in the resulting text were matched heuristically to the entities in the untranslated text using word alignment methods. This works well for entities that translate correctly or change little in the translation process but can be limited by the translation system, specifically if the system translates entities incorrectly.

For RE in low-resource languages, there has been limited focus on building training and testing data efficiently. However, crosslingual transfer methods have been applied to improve RE models, such as using multilingual BERT (Nag et al., 2021). Universal dependencies and sequence-to-sequence approaches have also been employed for RE in low-resource languages (Taghizadeh and Faili, 2022). Finally, recent sequence-to-sequence approaches for English have focused on extracting both relations and relation types (Cabot and Navigli, 2021).

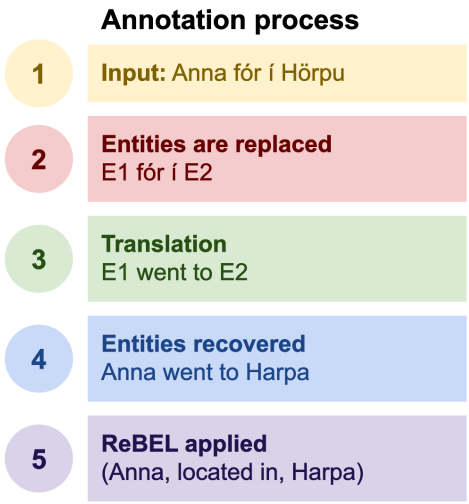


Figure 1: Summary of the annotation process for relation extraction studied in this paper.

4 Methodology

A general overview of the annotation approach is demonstrated in Figure 1. Below, we outline the

methods used in the process.

4.1 Models

For translation, we use the model from Facebook AI's WMT21 submission (Tran et al., 2021). The model is multilingual and has well-performing Icelandic to English translation capabilities.

For relation extraction, we use REBEL (Cabot and Navigli, 2021) a sequence-to-sequence relation extraction model that achieves a micro-F1 score of 75.4 on the well-studied CONLL4 corpus (Roth and Yih, 2004) and 93.4 on the NYT corpus (Riedel et al., 2010).

4.2 Data

For evaluating the precision of the relation extraction method, we use the first 200 sentences of each category from the MIM-GOLD-EL corpus (Friðriksdóttir et al., 2022). The number of unique and lemmatized entities for each category are shown in Table 5 in the Appendix.

We also performed a further evaluation on 200 sentences chosen uniformly at random from the sentences annotated above to get an estimate of the precision, recall and F1-score on the evaluated text for encoded vs. not encoded entities. We used strict evaluation as in (Taillé et al., 2020), and we additionally required that the name of the entity perfectly matched the lemmatized version of the entity's name in the knowledge base. We would like to emphasize that this annotation task is cognitively more demanding than solely estimating precision as no gold data exists for relation extraction in Icelandic. The task requires the annotator to find all relations in a given text, instead of just labelling the output of a model as correct/incorrect. As REBEL is an open relation extraction model and thus accounts for a very large amount of different relations, we restricted ourselves to those that had already appeared in our data (a total of 145 types).

4.3 Encoding Entities

We use gold-annotated entities. Each entity receives its own identifier and is replaced by it. The first occurring entity in the text receives the identifier E0, the next one E1, et cetera. For the sake of clarity, we note that if an entity appears multiple times in the text, it receives the same entity for all occurrences. This makes it clear which encoding refers to which original entity when decoded.

5 Results

Our results indicate that using the encoding method proposed can increase the number of correctly identified relations between two entities by up to 9.7% (Table 1). It is evident that a higher number of relation triplets is proposed by REBEL when the data is not encoded (Table 2) and we suspect this is due to the text being more fluently English, i.e. it contains less foreign (in our case, Icelandic) words within the English translation which should make it more natural for the monolingually English RE model. On the other hand, having a higher number of relation triplets proposed introduces significantly more noise, making the percentage of correctly identified relation triplets in fact lower. Additionally, encoding the data reduces the number of translation inconsistencies and errors. Conjugation of nouns tends to be mangled by the translation model, creating examples such as *Steingríms [genitive] Davíðsson [nominative]*, accents get dropped (Ása becomes Asa) and Icelandic letters modified (Þorgrímur becomes Thorgrímur). Summary statistics can be seen in Table 4 and by category in Table 5 in the Appendix.

It's worth noting that the text categories from our corpora that contain news have the highest precision scores and improvements using our method. This is not surprising since they were translated using WMT21's newsdomain parameter which should make them better translated. All categories score higher in correctly identified relation triplets when using the encoding method except one, the adjudications category. This is likely because there are few entities reported in that category and most of them are anonymized, and not counted as a relation because an anonymous person A being related to an anonymous person B is not informative for a knowledge base.

When looking at the overall performance of our model in Table 3 we observe higher recall than precision, which is explained by the high number of relations reported by the model. Having a higher preference for recall than precision is of great importance in a labelling task such as this one since it is generally cognitively less demanding to label the output of the model as correct/incorrect rather than to identify the missing relations in the text. However, the task becomes longer with lower precision since most of the reported relations will be irrelevant. When using

encoded entities we observed a big difference for relation triplets where both items in the relation consisted of entities, 11.5% F1-score for encoded entities vs. 1.9% for non-encoding. When evaluating the correctness of all relations (i.e., not only those between gold labelled entities), we saw a slight drop in F1-scores using our encoding method. This is not surprising, since the encoding method is intended to improve the extraction of relations between gold-annotated entities, which could come at the cost of performance for extracting other relations.

Category	Not	Encoded
Adjudications	3.4%	3.1%
Blog	1.0%	1.6%
Books	1.5%	5.3%
Emails	0.3%	5.5%
Newspaper 1	1.6%	8.9%
Newspaper 2	1.5%	7.1%
Laws	0.0%	0.5%
Radio/TV news	0.3%	7.6%
School essays	2.7%	4.0%
Scienceweb	1.3%	6.4%
Webmedia	2.1%	11.8%
Websites	1.3%	9.6%
Written to be spoken	1.9%	7.9%

Table 1: Precision score per category for all relations between two established entities. The reported numbers are the percentage of relation triplets labelled as correct per text category within our corpus.

	Not	Encoded
Total relations extracted	8910	8870
Entities in translation	3476	4368
Correct relations	2450	2078
Correct with entities	135	541

Table 2: Aggregated results on the data in Table 1. Entities in translation refers to the number of times that entities from MIM-GOLD-EL appear lemmatized in the resulting translation. Correct relations refers to the total number of relations labelled as correct, regardless of whether or not they contain established entities. Correct with entities refers to relation triplets that contain established entities as both head and tail.

5.1 Qualitative Evaluation of Errors

We note that while technically correct, the machine translation model tends to give various different translations to a single entity which certainly influences the higher number of unique relation triplets proposed by the RE model when working with data that has not been encoded. For instance, the Icelandic political party Samfylkingin gets translated in four different ways (Confederation, Alliance, Social democrats and Social democratic party) as well as the rescue worker association Landsbjörg (translated as either accident insurance company, accident prevention association, emergency rescue association or accident prevention society). Encoding the data avoids the problem of having to backtrack the translations in order to figure out whether or not they refer to the same original entity.

Using the encoding method, we can additionally ensure that all extracted relation triplets contain entities that include the entire, lemmatized mention of the entity. As per Icelandic naming conventions, a person is generally only referred to by their full name the first time they are mentioned in a given text and afterwards only referred to by their first name. When working with data that has not been encoded, we therefore get relation triplets that include only the first name of a person, potentially conjugated, while the encoded data always ensures that the person's entire, lemmatized name is present within the triple. This creates more consistency and avoids ambiguity in the output data.

It should, however, be noted that REBEL itself occasionally jumbles entity mentions itself even though the data has been encoded. Examples of this include when REBEL proposes that a person is a part of a family by that person's last name (i.e. *(Ingibjörg Sólrún Gísladóttir, is a member of, Gísladóttir)* but this is not how things work in Icelandic where patronyms are used instead of traditional last names. Another example is when REBEL adds international endings to websites that already include their Icelandic endings (such as when *tonlist.is* becomes *tonlist.is.com*).

6 Discussion

In this work, our focus was on evaluating an encoding method that can lead to improved automated relation extraction in text such as Icelandic. To eliminate any errors due to the recognition of entities, we based this evaluation on gold-

Method	Precision	Recall	F1-score	Evaluation
Encoded entities	26.1%	50.8%	34.5%	All relations
	6.3%	65.2%	11.5%	Between two entities
No encoding	26.9%	62.0%	37.5%	All relations
	1.0%	30.4%	1.9%	Between two entities

Table 3: Precision, recall and F1-scores for a subset of 200 examples chosen at random to estimate false negatives and hence recall and F1-score.

annotated entities. However, for labelling relations, our method can be combined with existing NER models such as from (Snæbjarnarson et al., 2022) that achieves an accuracy of 98.8% for Icelandic. We further believe that it would be interesting to study our approach for relation extraction methods that only report possible relations between entities instead of all possible relations in the text, i.e., where the entities can be provided as input to the relation extraction model. For further study of efficiency, it could be interesting to compare this method to heuristics that match entities in translated text.

One limitation of our study is the strict evaluation approach. We deemed the output of REBEL to be incorrect if the entities were not in their lemmatized form, shortened or otherwise modified. For example, when talking about someone using their first name only, REBEL does not have sufficient context to disambiguate the entity for insertion into a knowledge-base. This could be addressed by first disambiguating the entities before the text is processed in this manner. We did not evaluate how much the performance could increase, but we believe that a good disambiguation model could have a significant effect on the result.

Our approach addresses the low-performance of modern machine translation systems in translating entities correctly. Therefore, we would expect that improvements in machine translation would make our approach obsolete. However, that would require better translations of named entities. Unfortunately, for Icelandic, we do not have a corpus of entities translated to other languages such as English. Transliteration of named entities is the process of translating entities across languages and has been performed for English and several other languages (Grundkiewicz and Heafield, 2018). Transliterating named entities could be an approach to improve machine translation for Icelandic and would possibly make it more reliable to translate without any modifications to the source text and use word alignment to match entities between the source text and the translated text.

For creating data on relation extraction, we use machine-translation as an aid. However, to build a good relation extraction system for Icelandic, it might not be necessary to fine-tune the system on Icelandic. As an example, multilingual QA systems have shown good performance on Icelandic although they were not fine-tuned in QA for the language (Snæbjarnarson and Einarsson, 2022). We expect to see similar results for Icelandic and the data from this work can serve as a test set to measure the performance.

7 Conclusion

In conclusion, the proposed encoding method shows great potential for LRL, improving the percentage of correctly identified relations between entities by up to 9.7% for various categories of text. The method is simple and does not require any additional cost, making it ideal for languages where data is scarce and budget is limited. We note that this method can only be as good as the quality of the machine translation models as well as the RE methods for higher resourced languages. However, the encoding avoids several issues introduced by the bootstrapping method, making it more efficient with minimal effort.

Acknowledgments

This research was conducted with funding from The Strategic Research and Development Programme for Language Technology. We thank the anonymous reviewers for providing helpful comments on this manuscript and Valdimar Ágúst Eggertsson for helpful discussions.

References

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam Hruschka, and Tom Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 24, pages 1306–1313.

Sandipan Dandapat and Andy Way. 2016. Improved named entity recognition using machine translation-based cross-lingual information. *Computación y Sistemas*, 20(3):495–504.

Steinunn Rut Friðriksdóttir, Valdimar Ágúst Eggertsson, Benedikt Geir Jóhannesson, Hjalti Daníelsson, Hrafn Loftsson, and Hafsteinn Einarsson. 2022. Building an Icelandic Entity Linking Corpus. In *Proceedings of the workshop 'Dataset Creation for Lower-Resourced Languages', at the 13th International Conference on Language Resources and Evaluation (LREC 2022).*, Marseille, France.

Roman Grundkiewicz and Kenneth Heafield. 2018. Neural machine translation techniques for named entity transliteration. In *Proceedings of the Seventh Named Entities Workshop*, pages 89–94, Melbourne, Australia. Association for Computational Linguistics.

Alankar Jain, Bhargavi Paranjape, and Zachary C. Lipton. 2019. Entity projection via machine translation for cross-lingual NER. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1083–1092, Hong Kong, China. Association for Computational Linguistics.

Jing Jiang and ChengXiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 113–120, Rochester, New York. Association for Computational Linguistics.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 178–181, Barcelona, Spain. Association for Computational Linguistics.

ChunYang Liu, WenBo Sun, WenHan Chao, and Wanxiang Che. 2013. Convolution neural network for relation extraction. In *Advanced Data Mining and Applications: 9th International Conference, ADMA 2013, Hangzhou, China, December 14-16, 2013, Proceedings, Part II 9*, pages 231–242. Springer.

Arijit Nag, Bidisha Samanta, Animesh Mukherjee, Niloy Ganguly, and Soumen Chakrabarti. 2021. A data bootstrapping recipe for low-resource multilingual relation classification. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 575–587, Online. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Cícero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 626–634, Beijing, China. Association for Computational Linguistics.

Vésteinn Snæbjarnarson, Haukur Barri Símonarson, Pétur Orri Ragnarsson, Svanhvít Lilja Ingólfsdóttir, Haukur Jónsson, Vilhjalmur Thorsteinsson, and Hafsteinn Einarsson. 2022. A warm start and a clean crawled corpus - a recipe for good language models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4356–4366, Marseille, France. European Language Resources Association.

Vésteinn Snæbjarnarson and Hafsteinn Einarsson. 2022. Cross-Lingual QA as a Stepping Stone for Monolingual Open QA in Icelandic. In *Proceedings of the Workshop on Multilingual Information Access (MIA)*, pages 29–36, Seattle, USA. Association for Computational Linguistics.

Stephen Soderland, David Fisher, Jonathan Aseltine, and Wendy Lehnert. 1995. Crystal inducing a conceptual dictionary. In *Proceedings of the 14th international joint conference on Artificial intelligence-Volume 2*, pages 1314–1319.

Nasrin Taghizadeh and Heshaam Faili. 2022. Cross-lingual transfer learning for relation extraction using universal dependencies. *Computer Speech & Language*, 71:101265.

Bruno Taillé, Vincent Guigue, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. Let's Stop Incorrect Comparisons in End-to-end Relation Extraction! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3689–3701, Online. Association for Computational Linguistics.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook AI's WMT21 news translation task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 205–215, Online. Association for Computational Linguistics.

Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. 2016. Improved relation classification by deep recurrent neural networks with data augmentation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1461–1470, Osaka, Japan. The COLING 2016 Organizing Committee.

Yu Yan, Haolin Sun, and Jie Liu. 2021. A review and outlook for relation extraction. In *Proceedings of the 5th International Conference on Computer Science and Application Engineering*, CSAE '21, New York, NY, USA. Association for Computing Machinery.

A Summary Statistics

To highlight the scope of this work, we list important summary statistics in Table 4 and by category in Table 5.

	Encoded	**Not Encoded**
Total relations extracted	8870	8910
Unique triplets	1558	1805
Unique relation types	128	141

Table 4: Results on the overall data. Total relations refers to the total number of relations retrieved by REBEL. Unique triplets refer to the total number of unique relation triplets.

Category	**# Unique Entities**	# Not encoded	# Encoded
Adjudications	50	731	734
Blog	75	689	757
Books	118	752	740
Emails	89	355	343
Newspaper 1	207	696	676
Newspaper 2	213	722	690
Laws	96	694	651
Radio/TV news	132	708	686
School essays	89	747	753
Scienceweb	180	707	702
Webmedia	202	712	729
Websites	184	699	678
Written to be sp.	184	728	731

Table 5: The first column shows the number of unique, lemmatized entities in the first 200 sentences of each category. The second column depicts the number of relations where the data has not been encoded and the third the number of relations where the data has been encoded.

Improving Generalization of Norwegian ASR with Limited Linguistic Resources

Per Erik Solberg
National Library of Norway
Oslo, Norway
per.solberg@nb.no

Pablo Ortiz
Telenor Research
Oslo, Norway
pablo.ortiz@telenor.com

Phoebe Parsons[1] **Torbjørn Svendsen**[1] **Giampiero Salvi**[1,2]
1) Department of Electronic Systems, NTNU, Trondheim, Norway
2) KTH Royal Institute of Technology, EECS, Stockholm, Sweden
{phoebe.parsons,torbjorn.svendsen,giampiero.salvi}@ntnu.no

Abstract

With large amounts of training data, it is possible to train ASR models that generalize well across speakers and domains. But how do you train robust models when there is a limited amount of available training data? In the experiments reported here, we fine-tuned a pre-trained wav2vec2 ASR model on two transcribed, Norwegian speech datasets, one with parliamentary speech and one with radio recordings, as well as on combinations of the two datasets. We subsequently tested these models on different test sets with planned and unplanned speech and with speakers of various dialects. Our results show that models trained on combinations of the two datasets generalize better to new data than the single-dataset models, even when the length of the training data is the same. Our lexical analysis sheds light on the type of mistakes made by the models and on the importance of consistent standardization when training combined models of this kind.

1 Introduction

Automatic speech recognition has experienced tremendous development in the past decades. New models have improved in sophistication as well as complexity, see e.g., (Chorowski et al., 2015; Chan et al., 2015; Amodei et al., 2016; Synnaeve et al., 2019; J. Li et al., 2020; Baevski et al., 2020; B. Li et al., 2021; Radford et al., 2022) and references therein. However, these models are still at best as good as the data they are trained on. Spoken language presents many dimensions, such as degree of spontaneity, dialectal variation, task domain, and age of speaker, that affect the performance of a speech recognizer. Models trained on a specific combination of those factors will perform poorly in other conditions.

With sufficient computing capacity and data, this problem can be mitigated by training large models on even larger sets of data spanning several conditions, both in terms of the spoken content as well as transcription standards (e.g. Whisper, Radford et al., 2022). However, when computing power and quantity of data are limited, it is necessary to provide a dataset with a common transcription standard and enough variety of acoustic and linguistic features to fully encapsulate the target language. Such a dataset can then be used to fine-tune a large pre-trained model such as wav2vec2 (Baevski et al., 2020); a process less computationally expensive than training from scratch.

Until recently, freely available speech datasets suitable for training Norwegian ASR models only contained manuscript-read speech. However, in 2021, the Norwegian Language Bank released the Norwegian Parliamentary Speech Corpus (NPSC), containing 126 hours of transcribed speech from Stortinget, the Norwegian Parliament. The dataset contains a decent amount of spontaneous speech, as well as dialectal variation. Solberg and Ortiz (2022) showed that ASR models trained on the NPSC were better at transcribing spontaneous speech and dialects than models trained on manuscript-read speech only. Researchers at the National Library of Norway have fine-tuned wav2vec2 ASR models on the NPSC training split. Their 1B parameter model[1] obtained a word error

[1] https://huggingface.co/NbAiLab/nb-wav2vec2-1b-bokmaal

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 508–517

rate (WER) of 6.4% on the NPSC test split (De la Rosa et al., 2023).[2]

However, when we apply ASR systems trained on the NPSC to speech data such as broadcasts and informal speech, they tend to perform less well than on parliamentary speech. This is understandable since parliamentary speech is rather formal and contains few interruptions and instances of real conversation. Moreover, the topics discussed in Parliament are relatively restricted, and words which occur frequently in everyday conversations may be rare or nonexistent in such a dataset. Finally, the recording conditions are very homogeneous.

In this paper, we investigate the effect of finetuning wav2vec2 models on more varied Norwegian training data. Two models are trained on single training sets with unplanned speech, the NPSC and the broadcasting dataset Rundkast, while two others are trained on combinations of these datasets. We show that the models trained on combined data sources generalize better to new data than the models trained on single datasets and that this effect cannot be attributed to the length of the training data. Our analyses also show that *standardization*, the process of making transcriptions as uniform as possible across datasets, is key when combining training data in this way.

The outline of the rest of the paper is as follows. In Section 2, we describe the datasets used in the experiments. Section 3 reports on how the models were trained, as well as the experimental setup. The results of the experiments are described and discussed in Section 4, and we also use a technique from corpus linguistics called *keyword analysis* to get a deeper understanding of the differences between the models. Section 5 concludes the paper and suggests some avenues for further development and research, based on our results and the state of Norwegian ASR.[3]

2 Datasets for Norwegian ASR

There are a number of datasets that can be used for training and testing Norwegian ASR systems. This section describes the most important, freely accessible Norwegian speech datasets, as well as one which is not freely accessible, Rundkast. All of these datasets have transcriptions in Norwegian Bokmål, the most commonly used of the two written standards of Norwegian, while some also have transcriptions in Nynorsk. In the experiments of this paper, we only use the Bokmål transcriptions.

2.1 NST

The now defunct company Nordisk språkteknologi ('Nordic Language Technology'; NST) made a large dataset for training and testing their ASR system in the late 90s and the beginning of the millennium. This dataset has been shared with an open license at the Norwegian Language Bank at the National Library of Norway since 2011.[4] The NST dataset includes around 540 hours of manuscript-read speech in Bokmål from close to 1000 informants. They read sentences from newspapers, but also some repeated words and sequences of numbers. Since the dataset only contains planned speech, there are few instances of hesitations, repetitions, false starts, etc., which are more frequent in unplanned speech. The speakers come from different dialect areas, but since the speech is scripted, the speech deviates less from the Bokmål norm than unscripted speech. This accounts for why models trained only on the NST generalize less well to different dialects than systems trained on both NST and the NPSC in (Solberg and Ortiz, 2022).

2.2 NPSC

The NPSC was developed by the Norwegian Language Bank in 2019-2021 as an open dataset for ASR of Norwegian unscripted speech (Solberg and Ortiz, 2022).[5] This dataset consists of about 126 hours[6] of recordings of meetings from 2017 and 2018 at the Norwegian Parliament. These are transcribed manually by trained linguists. There are transcriptions both in Bokmål (87%) and Nynorsk (13%). Individual speakers are transcribed consistently in one or the other written standard, following the practice in the official parliamentary proceedings. There are different versions of the transcriptions intended for different use cases (cf. Solberg and Ortiz, 2022,

[2]The data processing and training setup for those models are somewhat different from the NPSC-trained model in the experiments reported in this paper, and thus the results slightly differ.

[3]Code for the experiments is available at `https://github.com/scribe-project/nodalida_2023_combined_training`.

[4]`https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-54/`

[5]`https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-58/`

[6]Excluding breaks. The total duration is 140 hours.

sect. 2.2). We use the version in which numbers, dates and years are written with letters instead of digits, and abbreviations are not used. The NPSC also contains metadata about non-standard and/or dialectal words, which we use to standardize transcriptions, as described in subsection 2.5. There are 267 speakers in the dataset.

2.3 NB Tale

NB Tale is also an open speech dataset from the Norwegian Language Bank, published in 2015.[7] It is divided into three modules, two of which are used in this paper: Module 1 consists of manuscript-read sentences from newspapers by native speakers of different Norwegian dialects. The sentences are chosen to cover as many phonological phenomena as possible and are transcribed both orthographically and phonetically. Only the orthographic transcriptions are used in the experiments reported here. Some of the sentences in the dataset are read by all speakers, while others are read by a subset of the speakers or only one speaker. There are detailed metadata about each speaker, including dialect, age, and gender. Module 3 consists of recordings of the same speakers as module 1, as well as some non-native speakers (excluded from our analyses), speaking freely for 2 minutes on a subject of their choice. These are orthographically transcribed. There are 380 speakers in NB Tale. Module 3 is Bokmål only. 14.2 % of module 1 is in Nynorsk.

2.4 Rundkast

Rundkast, the only one of these datasets which does not have an open license, was developed by the Norwegian University of Science and Technology in 2005-2006 (Amdal et al., 2008). It consists of 77 hours of orthographically transcribed radio news and actuality shows from NRK, the Norwegian state broadcaster.[8] The written standard of the transcriptions is either Bokmål (80%) or Nynorsk (12%), depending on the dialect of the speaker.[9] Only the Bokmål transcriptions are used here. The dataset includes read news, interviews, debates, and commentary.[10]

[7] https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-31/

[8] A small subset is also phonetically transcribed.

[9] 8% are tagged as neither.

[10] Due to inconsistent uses of speaker names in Rundkast, it is not possible to make a reliable speaker count for this dataset.

2.5 Standardization and usage of combined data

Naturally, the four datasets described above have different transcription standards and metadata. We provide a set of standardizing procedures that aim to unify transcriptions from the aforementioned corpora such that they can be combined and used together consistently.[11] Of particular importance is the treatment of digits, abbreviations, non-verbal noises (e.g., hesitations), and non-standard and dialectal words across datasets. The most important changes to the original transcriptions are:

- Remove all special characters and punctuation except for "é" and "-", which appear often in Norwegian and can make a difference in the words' meaning.
- Substitute all the digits by letters according to how they are pronounced. While numbers, years and dates are written with letters in the datasets used here, the original transcriptions of the NPSC include some company names etc. which contain digits.
- Substitute non-verbal noises by three variants: "mmm" (nasal hesitation), "eee" (vowel hesitation), and "qqq" (other non-language vocal sounds such as laughter or coughing).
- Non-standard words and dialectal words are by default not standardized, for the purpose of having orthographic transcriptions that reflect as close as possible what is actually said. As a consequence, the transcriptions may contain words that are not in standard dictionaries of Bokmål.

The train, test and validation splits are performed on NST, NPSC and Rundkast individually, while NB Tale modules 1 and 3 are used for testing purposes only. For NPSC, we use the official splits. Parliamentary meetings are used as the minimum unit, i.e. they are not divided across different splits. This is to minimize the overlap in topics and vocabulary across splits (Solberg and Ortiz, 2022). For the NST, there was only an official train and test split. We, therefore, split the official test set into a test and validation set randomly. For Rundkast we performed a split using full programs

[11] The code that implements all the procedures described in this section, as well as the procedures for creating data splits, is available at https://github.com/scribe-project/asr-standardized-combined

as the minimum unit, and allocate programs with the largest number of different speakers to the test set, and then to the validation set. This is to better evaluate the generalization capabilities of the models. The Rundkast splits were kept as close as possible to the proportion 80:10:10 in terms of duration of the train, test and validation sets, respectively. These are also the proportions of the official NPSC splits (Solberg and Ortiz, 2022, sect. 3.4).

3 Experiments

The goal of our experiments is to verify to what extent we get improvements in performance when training on more varied spontaneous data than the NPSC data alone. To this aim we fine-tuned wav2vec2 models on both Rundkast and the NPSC individually, as well as on combinations of the two datasets. We then test the models on test sets from the same domain as the training set, as well as from different domains using the NB Tale and the NST corpora.

3.1 ASR framework and hyperparameters

All models described in Section 3.2 are based on the wav2vec2 architecture (Baevski et al., 2020). Inspired by the fine-tuning of Norwegian 300M parameter models in (De la Rosa et al., 2023), we used the Swedish 300M parameter wav2vec2 model trained in Sweden by Kungliga Biblioteket[12] (Malmsten, Haffenden, and Börjeson, 2022) as a starting point, and fine-tuned it to different sets of data. All training sessions used the default hyper-parameters in Huggingface transformer implementation, with the exception of the initial learning rate for the Adam optimizer that was set to 10^{-4}. All models were trained for 30 epochs, and the checkpoint with the lowest WER on the validation set was chosen for the recognition experiments on the test set.

Recognition was performed in two different settings for each model (without and with language model). In the first, the most likely token for each time step is first computed based on the output activations of the model. The sequence of best tokens is then passed to the tokenizer for decoding into words. In the second setting, the output activations of the model are passed directly to the tokenizer that uses beam search and a language model to produce the textual output. In this setting, we used the 5-gram model produced by researchers at the National Library of Norway[13].

As a reference we also performed recognition on our test sets with the large Whisper model (1.55 billion parameters) trained on a total of 680000 hours of (multilingual) speech, including 266 hours of Norwegian. In this case, the model is used without fine-tuning. When computing word error rates, in this case, we used Whisper's 'basic' text normalizer, followed by normalization of most numerals to minimize the discrepancies between reference text and the Whisper transcriptions. However, the corresponding results are not directly comparable with the wav2vec2 results because Whisper is trained to produce a loose transcription of speech rather than word-by-word transcriptions. Those results will therefore only be used for discussion. It is worth noting that both wav2vec2 and Whisper can output any sequence of characters (not only words out of a fixed vocabulary). For this reason, it is interesting to analyze not only the number of word errors, but also spelling variants or mistakes. This will be done in details in Section 4.3.

3.2 Models

We fine-tuned four models on four different training sets.[14] To distinguish models from training sets, model names are written in small caps. The model trained on the NPSC only is called STORTINGET, the name of the Norwegian Parliament.

RADIO The RADIO model is trained on Bokmål segments from the Rundkast training set with a segment length above 1 second and below 15 seconds. This amounts to 43.6 hours of audio.

STORTINGET The STORTINGET model is trained on Bokmål segments from the NPSC training set with a segment length above 1 second and below 15 seconds, which adds up to 70.3 hours of audio. That is, 80.4% of the original Bokmål training set.

COMBINED SHORT The COMBINED SHORT model is trained on a random sample of segments from the training sets of RADIO

[12] https://huggingface.co/KBLab/wav2vec2-large-voxrex

[13] https://huggingface.co/NbAiLab/nb-wav2vec2-kenlm

[14] These models are published on Huggingface: https://huggingface.co/scribe-project.

and STORTINGET (half of the total duration comes from each dataset). The total duration of the training data for COMBINED SHORT is 70.4 hours, only 4 minutes longer than the training set of the STORTINGET model. Thus, we cannot attribute performance differences between the two models to the size of training data.

COMBINED LONG Finally, the COMBINED LONG model is trained on the combination of the training sets of RADIO and STORTINGET without leaving anything out, which amounts to 114 hours.

3.3 Test Sets

The models were tested on NB Tale modules 1 and 3 and the test sets of the NPSC, Rundkast and NST. In all these datasets, we have filtered out segments shorter than one second and longer than 15 seconds.

Table 1 gives an overview of the duration, domain and use of the dataset samples used in the experiments.

4 Results and Analyses

4.1 Results per dataset

Table 2 reports the WER per dataset, for each model. When looking at the results, it makes sense to inspect the NPSC and Rundkast test sets separately from the others, since those are the test sets of the models' training data. As expected, the RADIO model outperforms the STORTINGET model on the Rundkast test set (17.8% vs. 24.0%), and conversely, the STORTINGET model outperforms the RADIO model on the NPSC test set (9.3% vs. 19.5%). The COMBINED SHORT model has a slightly better WER than the RADIO model (17.2%) on the Rundkast test set. This is not necessarily surprising, given that the COMBINED SHORT model is trained on a larger dataset than the RADIO model. Still, COMBINED SHORT is trained on less Rundkast data than RADIO, and adding data from a different domain seems to have a positive effect. The COMBINED LONG model gives the best score on the Rundkast test set (15.9%). For the NPSC test set, COMBINED SHORT does not improve on STORTINGET (9.9% vs. 9.3%). Again, COMBINED LONG has the lowest score (7.9%).

COMBINED LONG is the best model on the datasets which are not the models' test sets: NB Tale modules 1 and 3 and NST test. Furthermore, COMBINED SHORT has a better WER than both non-combined models. Since COMBINED SHORT is trained on the same amount of data as STORTINGET, this effect cannot be attributed to the size of the training set, so the improved generalization of the model seems to be due to the mixing of datasets. It is interesting to observe that RADIO performs better than STORTINGET on NB Tale 1 (24.5% vs. 25.8%) and on NST (10.6% vs. 11.2%). RADIO performs worse than STORTINGET on NB Tale 3, however (28.9% vs. 25.8%). NB Tale 1 and NST are manuscript-read datasets, while NB Tale 3 is spontaneous, which may be part of the explanation for the difference. It is not clear why, though, as neither STORTINGET nor RADIO are primarily manuscript-read.

When we look at the results with a language model, we see the same general trends, but with lower WER values: COMBINED SHORT is consistently better than STORTINGET on all datasets, with the exception of the NPSC test.

In addition to the limited resource models, we also tested the 1.55 billion parameter Whisper model (Radford et al., 2022) which is trained on massive amounts of heterogeneous data, from different sources, task domains and in multiple languages. As previously mentioned, the results from Whisper are not directly comparable, because the style of transcriptions from this model is different. However, the Whisper performance on NB Tale 1 and 3 and on NST is as good or better than the fine-tuned models (17.5%, 22.6% and 9.5% WER respectively). On the other hand, Whisper's performance on NPSC, Rundkast is clearly lower than our fine-tuned models (16.6% and 25.0% WER respectively). A possible explanation is that data from the NB Tale and NST datasets is not included in the training material for fine-tuning, reducing the effect of fine-tuning for these test sets, while Whisper benefits from the much larger and heterogenous training data.

4.2 Results per dialect

We report results per dialect in Table 3. Here we focus on NB Tale module 3 only because this part of the dataset contains spontaneous speech by a balanced set of dialect speakers, and is therefore well-suited for dialect-wise testing.

The dialect region *east* includes the Oslo re-

Dataset	train	test	validation	Domain	Use
NPSC	70.3h	9.1h	9.6h	mixed	train/test/validation
Rundkast	43.6h	5.9h	5.5h	mixed	train/test/validation
NST	n/a	25.6h	n/a	read	test
NB Tale 1	n/a	9.3h	n/a	read	test
NB Tale 3	n/a	7.4h	n/a	spontaneous	test
Combined short	70.4h	n/a	9.7h	mixed	train/validation
Combined long	114.0h	n/a	15.1h	mixed	train/validation

Table 1: Overview for the datasets samples used in the experiments.

Model	(training hours)	NPSC	Rundkast	NST	NBT1	NBT3
RADIO	(43.6h)	19.5 (14.9)	17.8 (15.3)	10.6 (7.1)	24.5 (19.5)	28.9 (23.2)
STORTINGET	(70.3h)	9.3 (8.1)	24.0 (20.5)	11.2 (7.8)	25.8 (20.7)	25.8 (21.5)
COMBINED SHORT	(70.4h)	9.9 (8.3)	17.2 (14.6)	9.2 (6.3)	23.5 (19.1)	24.4 (19.9)
COMBINED LONG	(114.0h)	**7.9** (**7.1**)	**15.9** (**14.0**)	**8.6** (**6.0**)	**21.9** (**18.0**)	**23.0** (**19.2**)

Table 2: Word error rates (%) per model and test set. The best results with limited linguistic resources (wav2vec2) are shown in **bold** and the second best are underlined. Results in parentheses are obtained by combining the wav2vec2 models with a 5-gram language model.

gion and the counties in the southeastern part of Norway. *South* groups together the dialects in the county of Agder on the south coast. *West* includes the dialects on the southwest coast of the country in the counties of Rogaland, Vestland, and southern Møre og Romsdal. *Mid* includes the dialects in the county of Trøndelag and northern Møre og Romsdal, while *north* covers the dialects north of Trøndelag, in the counties of Nordland and Troms og Finnmark.

As we saw in the previous section, RADIO has poorer performance than STORTINGET on NB Tale module 3 globally, and we see that this difference is reflected in all dialect regions.

All models perform best on the eastern dialects. This is not surprising, as more than half of the population lives in this region (Thorsnæs, 2023) and there is a bias in the models' training data towards the dialects in this region. Moreover, many of the eastern dialects, in particular those in the Oslo region, are close to the written Bokmål norm.

From Table 3 we can see that all models struggle most with the *mid* and *west* dialects. Many dialects from these areas have inflections and lexical forms of words which differ substantially from Bokmål. Moreover, the models are exposed to limited amounts of western Norwegian, as many of the speakers from the west coast are transcribed in Nynorsk in Rundkast and the NPSC. Nynorsk transcriptions are filtered out in the datasets used for training and testing these models.

COMBINED SHORT improves on the single dataset models for all dialect regions. The improvement from RADIO to COMBINED SHORT is substantial across regions, ranging from a relative improvement of 13.5% for *east* to 17.9% for *south*. Again, this is not surprising, as COMBINED SHORT is trained on more data. From STORTINGET to COMBINED SHORT, there are also improvements, although less substantial: COMBINED SHORT improves on STORTINGET by a relative 10.2% for *east*, while for the *mid* region, the WER is almost the same for the two models. For the other regions, the relative improvements are below 7%.

4.3 Lexical analysis

To better understand the kinds of errors the models make we have used a technique from corpus linguistics called *keyword analysis* (Dunning, 1994; Pojanapunya and Todd, 2018, and references therein). Keywords are words that have a surprising frequency, either surprisingly high or surprisingly low, in a target corpus relative to a reference corpus. Words are assigned a value indicating their keyness. Two common statistics used to compute keyness are log-likelihood (LL) and χ^2. In this study, we will use LL, which is the more reliable statistics when the expected frequency of

Model	east	west	mid	north	south
RADIO	22.3	32.7	32.0	27.1	25.7
STORTINGET	21.5	28.6	27.7	24.6	22.3
COMBINED SHORT	<u>19.3</u>	<u>27.2</u>	<u>27.4</u>	<u>22.9</u>	<u>21.1</u>
COMBINED LONG	**18.2**	**25.8**	**25.5**	**21.5**	**20.5**

Table 3: Word error rates (%) per dialect in the NB Tale module 3 test set. The best results are shown in **bold** and the second best are <u>underlined</u>.

a word is low (Dunning, 1994).

Keyword analysis is often used to characterize a text's genre or identify its ideological underpinnings. It can also be used to generate term lists for a given field or topic (Pojanapunya and Todd, 2018). In this study, we used keyword analysis to identify word forms that characterize a machine transcription relative to the ground truth. For each machine transcription, we looked through the list of the 100 words with the highest LL value. Such a keyword list can reveal the words contributing to the machine transcription WER. Words that have either an unusually high or an unusually low relative frequency in the machine transcription relative to the ground truth, will get a high LL and will therefore appear in the keyword lists. Both cases may reveal properties of the transcription. One limitation of this method is that word forms occurring only once in the target corpus and never in the reference corpus will get a low LL and not appear in the keyword list. Therefore, we may miss misspellings.

There are many instances of incorrectly spelled words with a high frequency in the automatic transcription and a zero frequency in the ground truth. Often, the correctly spelled version is also present in the list, with a higher frequency in the ground truth than in the automatic transcription. Typically, these words have a spelling that is surprising given the pronunciation of the word, such as foreign company names and loan words. The genitive of Apple, "apples" has a frequency of 241 in the ground truth and 0 in the STORTINGET transcription.[15] The STORTINGET list contains misspellings of this name, only occurring in the automatic transcription, such as "appels" and "apels". Similarly, "rock" occurs 37 times in the ground truth, but never in the ASR output from STORTINGET.

We see a similar phenomenon with uncommon words. As mentioned, the sentences in NB Tale module 1 are chosen to cover as many phonological phenomena as possible, and many sentences are repeated by several or all informants. As a consequence, there are quite a few uncommon words in that dataset, and some of them appear in the keyword analysis. An example is the word "stokkmaur" ('Carpenter ant'), which occurs 240 times in the ground truth, but only 56 times in the STORTINGET transcriptions. The STORTINGET list contains two misspellings of this word, however: "stokmaur", "stokkmør". RADIO prefers to spell this word "stockmaur".

There are quite a few examples of words where a vowel is left out, possibly due to fast speech, such as "tittlen", instead of "tittelen" ('the title'). However, there are not many obvious examples of dialect pronunciations, except for some very frequent function words. This does not necessarily mean that dialectal pronunciations do not contribute to the WER of the models: Dialectal transcriptions could be hapaces, forms occurring only once, which will not get a high LL value. The ground truth transcriptions of native speakers has 9672 hapaces, while hapax count for STORTINGET and RADIO is more than twice as high. They have 20591 and 20215 hapaces respectively. The hapax count goes down to 18676 for COMBINED SHORT and 17613 for COMBINED LONG. An inspection of a sample of the hapaxes from the different models which don't also occur in the ground truth, reveal that they are, to a large extent, misspellings. It is, however, hard to tell from reading the misspellings whether they are of dialectal origin or not. The hapax count goes further down when a language model is used (STORTINGET: 13576, RUNDKAST: 13125, COMBINED SHORT: 12076, COMBINED LONG: 12065), which indicates that the language model

[15] All words are spelled with lowercase letters in the datasets. Both NB Tale module 1 and NST contain many repeated sentences, which likely accounts for the high frequency of this genitive form. Note also that genitive forms are spelled without an apostrophe in Norwegian.

Model	NPSC	Rundkast	NST	NBT1	NBT3
RADIO	17.4 (13.8)	16.0 (13.2)	9.7 (7.1)	22.5 (18.5)	27.2 (21.4)
STORTINGET	8.8 (7.4)	21.1 (17.0)	11.1 (7.8)	25.3 (19.9)	24.4 (19.5)
COMBINED SHORT	9.3 (7.6)	15.4 (12.5)	8.7 (6.3)	21.7 (18.1)	22.8 (18.2)
COMBINED LONG	**7.2** (**6.2**)	**14.1** (**11.9**)	**7.8** (**6.0**)	**19.5** (**17.0**)	**21.2** (**17.5**)

Table 4: Word error rates (%) per model and test set with hesitations removed and with standardization of compounds and acronyms. Best results are in **bold** and second best are underlined. Results in parenthesis are obtained combining the wav2vec2 model with a 5-gram language model.

reduces the number of spelling mistakes.

The test sets have special markings for hesitations, and the models are trained to produce such markings too. This appears to be a source of errors. In particular, nasal hesitations, marked as "mmm", occur 994 times in the ground truth, but almost never in the automatic transcriptions, and it is the word with the highest LL in the keyword analyses of all the models. This kind of error does not affect the semantics of the transcription, and markings of hesitations will presumably be removed in many downstream applications.

Another source of errors which does not impede the understanding of the transcriptions, is insufficient standardization of the different datasets. When comparing the analyses of the two single-dataset models, it turns out that STORTINGET tends to transcribe compounds without a hyphen, e.g. "arbeiderpartipolitikeren" ('the Labor Party politician'), while RADIO tends to use a hyphen: "arbeiderparti-politikeren". The datasets STORTINGET is trained on, the NPSC, also transcribes compounds without a hyphen while Runkast, which RADIO is trained on, uses a hyphen, and this difference is not captured by the standardization routines described in section 2.5. There is a similar issue with acronyms. STORTINGET transcribes acronyms such as NRK, the national broadcaster, as "nrk", while RADIO separates each letter with a space, "n r k", also due to a difference in the training data which is not captured by the standardization routines. Unsurprisingly, the combined models produce a mix of these standards.

While hesitations and differences in transcription standards contribute to the WER, they are in a sense less important than misspellings and wrong words, which may affect the comprehension and the usability of the transcriptions. We would, therefore, like to check to what extent these errors contribute to the WER. Can the higher performance of COMBINED SHORT compared to the single-dataset models be explained entirely by these errors? To check this we have made a version of the ground truth and the automatic transcriptions where hesitations are removed, compounds are written without hyphens, and where a number of the most frequent acronyms are written without a space between them. Table 4 reports the results across datasets with these standardizations. The values in parentheses are the WER with a language model. The results should be compared to those in table 2. As before, the COMBINED SHORT model outperforms the single dataset models on all datasets except the NPSC test set, where the STORTINGET model is better. To see the effect of this cleaning of hesitations, hyphens and spaces, we can look at the global WER across all the datasets (excluding foreign speakers). Before cleaning the global WER is 16.8% for STORTINGET, 18.1% for RADIO, and 14.7% for COMBINED SHORT without a language model. After cleaning, the global WERs are 16.0%, 16.5%, and 13.6% respectively. The relative improvement of the global WER from STORTINGET to COMBINED SHORT is 12.5% before cleaning and 15.0% after cleaning. For RADIO, the improvement is 18.8% before cleaning and 17.6% after cleaning. In other words, when we exclude the errors we have observed which stem from hesitation annotations or differences in transcription standard, the gap between STORTINGET and COMBINED SHORT becomes somewhat larger and the gap between RADIO and COMBINED SHORT becomes somewhat smaller, but COMBINED SHORT still improves on the single-dataset models. The difference between the single dataset models and the combined dataset models cannot be explained solely by the transcription of hesitations and the differences in transcription standards.

5 Conclusion and future work

In this paper, we have shown that training ASR models on a combination of parliament speech data from the NPSC and broadcast data from Rundkast results in better WER across different test sets than models trained on these datasets individually. This effect persisted even when we control for dataset length. While STORTINGET is slightly better on the NPSC test set than the combined model of similar length, COMBINED SHORT outperforms both single dataset models on all other test sets. In other words, the combined models generalize better to out-of-domain speech data, which makes them more suitable for downstream transcription tasks where different kinds of speech data may be encountered, such as meeting transcriptions and subtitling.

The study also highlights that it is important to standardize the training and test data when combining datasets in this way. This standardization may require an intimate knowledge of the transcription guidelines of the different datasets. Even though we had standardized the datasets prior to training, as described in section 2.5, we did not discover the differences in the treatment of acronyms and compounds until we investigated the ASR outputs in detail.

To be able to train on combinations of datasets, one needs to have access to ASR dataset of different types and genres. Before the release of the NPSC in 2021, there were no large, open datasets for ASR training with Norwegian unplanned speech. The NPSC may be released openly because parliamentary recordings are in the public domain. Due to copyright and privacy issues, it is more difficult to make a dataset with broadcast data such as Rundkast freely available. A recent report from the Norwegian Board of Technology points out that there are not enough open ASR datasets for Norwegian, and the datasets that exist are not sufficiently varied. It suggests different ways to increase the amount of open speech data, such as a major crowdsourcing initiative (Tennøe and Wettre, 2022). While we wait for more open data, it may be possible to train ASR models on a combination of open and non-open datasets and release the resulting models openly.

Finally, the results we obtained with Whisper are not comparable to ours using WER. This is because the model is trained to produce transcriptions of different standards. This emphasizes the importance of developing new metrics that assess the semantic content of the transcriptions rather than the word accuracy.

Acknowledgments

This work has been partially supported by the Research Council of Norway through the IKTPLUSS grant for the SCRIBE project[16] (KSP21PD).

References

Amdal, Ingunn, Ole Morten Strand, Jørgen Almberg, and Torbjørn Svendsen (2008). "RUNDKAST: an Annotated Norwegian Broadcast News Speech Corpus." In: *Proceedings of the 6th Conference on Language Resources and Evaluation*, pp. 1907–1913.

Amodei, Dario et al. (2016). "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin." In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, pp. 173–182.

Baevski, Alexei, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli (2020). "wav2vec 2.0: A framework for self-supervised learning of speech representations." In: *Advances in Neural Information Processing Systems* 33, pp. 12449–12460.

Chan, William, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals (2015). "Listen, Attend and Spell." In: *CoRR* abs/1508.01211. arXiv: 1508.01211. URL: http://arxiv.org/abs/1508.01211.

Chorowski, Jan, Dzmitry Bahdanau, Dmitriy Serdyuk, KyungHyun Cho, and Yoshua Bengio (2015). "Attention-Based Models for Speech Recognition." In: *CoRR* abs/1506.07503. arXiv: 1506.07503. URL: http://arxiv.org/abs/1506.07503.

De la Rosa, Javier, Rolv-Arild Braaten, Per E Kummervold, and Freddy Wetjen (2023). "Boosting Norwegian Automatic Speech Recognition." In: *Proceedings of the 24rd Nordic Conference on Computational Linguistics (NoDaLiDa)*. Faroe Islands: Linköping University Electronic Press, Sweden.

Dunning, Ted (1994). "Accurate methods for the statistics of surprise and coincidence." In: *Computational Linguistics* 19, pp. 61–74.

[16] https://scribe-project.github.io/

Li, Bo et al. (2021). "Scaling End-to-End Models for Large-Scale Multilingual ASR." In: *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1011–1018. DOI: 10.1109/ASRU51503.2021.9687871.

Li, Jinyu et al. (2020). "On the Comparison of Popular End-to-End Models for Large Scale Speech Recognition." In: *Proc. Interspeech 2020*, pp. 1–5. DOI: 10.21437/Interspeech.2020-2846.

Malmsten, Martin, Chris Haffenden, and Love Börjeson (2022). *Hearing voices at the National Library – a speech corpus and acoustic model for the Swedish language*. DOI: 10.48550/ARXIV.2205.03026. URL: https://arxiv.org/abs/2205.03026.

Pojanapunya, Punjaporn and Richard Watson Todd (2018). "Log-likelihood and odds ratio: Keyness statistics for different purposes of keyword analysis." In: *Corpus Linguistics and Linguistic Theory* 14, pp. 133–167.

Radford, Alec et al. (2022). *Robust Speech Recognition via Large-Scale Weak Supervision*.

Solberg, Per Erik and Pablo Ortiz (2022). "The Norwegian Parliamentary Speech Corpus." In: *Proceedings of the 13th Conference on Language Resources and Evaluation*, pp. 1003–1008.

Synnaeve, Gabriel et al. (2019). "End-to-end ASR: from Supervised to Semi-Supervised Learning with Modern Architectures." In: *CoRR* abs/1911.08460. arXiv: 1911.08460. URL: http://arxiv.org/abs/1911.08460.

Tennøe, Tore and Jonas Engestøl Wettre (2022). *Taleteknologi med kunstig intelligens*. Tech. rep. Norwegian Board of Technology.

Thorsnæs, Geir (2023). *Østlandet*. Accessed on January 26, 2023. URL: http://snl.no/%5C%C3%5C%98stlandet.

The Finer They Get: Combining Fine-Tuned Models For Better Semantic Change Detection

Wei Zhou
University of Stuttgart
Bosch Center for Artificial Intelligence
wei.zhou@ims.stuttgart.de

Nina Tahmasebi
University of Gothenburg
nina.tahmasebi@gu.se

Haim Dubossarsky
Queen Mary University of London
Cambridge University
h.dubossarsky@qmul.ac.uk

Abstract

In this work we investigate the hypothesis that enriching contextualized models using fine-tuning tasks can improve their capacity to detect lexical semantic change (LSC). We include tasks aimed to capture both low-level linguistic information like part-of-speech tagging, as well as higher-level (semantic) information.

Through a series of analyses we demonstrate that certain combinations of fine-tuning tasks, like sentiment, syntactic information, and logical inference, bring large improvements to standard LSC models that are based only on standard language modeling. We test on the binary classification and ranking tasks of SemEval-2020 Task 1 and evaluate using both permutation tests and under transfer-learning scenarios.

1 Introduction

The last few years have seen a growing interest in language change, specifically in lexical semantic change (LSC), from the NLP community. LSC is the linguistic phenomenon in which words' meaning may change over time (for example by adding senses, or broadening/narrowing in their meaning scope). Originally, the study of lexical semantic change aspired to understand the phenomenon from a linguistic perspective (Dubossarsky et al., 2016; Schlechtweg et al., 2017; Keidar et al., 2022). However, it was also motivated by the need for better handling of semantic change in other text-based research disciplines that work with historical texts (e.g., lexicographers, historians).

In more recent times, the understanding that general purpose NLP models also need to accommodate for the fluidity of word meaning has reached the greater NLP community, bringing with it the realization that LSC plays a vital role (Barbieri et al., 2022). It is particularly visible in the deterioration of model performance over time because the language on which models and other algorithms are (pre-)trained, starts to drift as time passes (Rosin and Radinsky, 2022; Alkhalifa et al., 2023). When deployed, these models process text from time periods they were not trained on, which hinder their performance.

This wide, multi-disciplinary interest in LSC, has driven the development of many computational models for language change (Kutuzov et al., 2018; Tahmasebi et al., 2021). In addition, much work has been devoted to supporting this progress by curating evaluation datasets that provide appropriate testing of these new models. Most of these datasets are in the form of SemEval tasks and contain high quality, humanly annotated lists of words. Each word has either changed its meaning between the considered time periods, or remained stable in meaning, and each list is accompanied by relevant historical corpora. This has become the de-facto evaluation standard in the field.

When reviewing the different models that are evaluated in SemEval, most of them use the same suit of methods that rely on standard distributional models of meaning. These models are either trained solely on historical text (e.g., SGNS or other static models), or use contextualized models pre-trained on a large "general purpose" text.[1] All of these models generate meaning representations in vector form for words from a historical corpus and compare them to vectors representing the modern meaning of the same word. Although the models differ (e.g., in terms of data or training objectives), they all share the same basic trait - they rely on meaning representation based on neighboring words without additional linguistic information.

[1] Some contextualized models are also fine-tuned on historical corpora.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 518–528

This state of affairs raises a question: do models that are based solely on collocation statistics and trained exclusively on a masked-word prediction task suffice for our purpose? That is, are classic distributional models able to capture the full repertoire of word meaning, and then access it when analyzing meaning change? In this work, we investigate several concerns that we think a-priori suggest that additional linguistic information is beneficial for unsupervised lexical semantic change detection.

One direct path forward is fine-tuning pre-trained models on additional tasks or domains like Question Answering and Sentiment Analysis. It has been demonstrated that fine-tuning of models helps even when the fine-tuned tasks are different than the target task (for example, fine-tuning on textual summarization and testing on Question-Answering) in a sort of a zero/few-shot transfer learning (Peters et al., 2019; Merchant et al., 2020; Khashabi et al., 2020). Therefore, it is reasonable to assume that enriching a contextualized model with additional fine-tuning tasks would lead to improved performance also for LSC detection.

In this paper, we test this hypothesis with one of the top-performing models of LSC detection in English and explore the potential to improve it by enriching it through a set of fine-tuning tasks. We provide our code here.[2]

2 Related literature

2.1 Models of LSC and their evaluation

In the past years, we have seen an increasing amount of models for unsupervised detection of lexical semantic change, almost exclusively focused on distributional semantic models. SemEval-2020 Task 1 was the first attempt at a large-scale evaluation and comparison of methods on four different languages. Two main classes of methods were evaluated by the participating teams. The first was based on type embeddings, either those that require alignment between independently trained models, (e.g., SGNS with Orthogonal Procrustes alignment (Arefyev and Zhikov, 2020)) or static embeddings without the requirement of alignment (Zamora-Reina and Bravo-Marquez, 2020). The second class was based on contextualized embeddings and combined with e.g., a clustering algorithm to derive sense information (e.g., XLM-R with K-means clustering (Gyllensten et al., 2020)) or other means of comparing vectors in each time period with each other (Kutuzov and Giulianelli, 2020). The models were evaluated on two tasks, binary classification and ranking.

For the four SemEval datasets, the trend was that type-based models outperformed contextualized (token) models. It was also clear that good performance on binary classification does not necessarily indicate good performance on the ranking task. Since SemEval-2020, more evaluation tasks have been curated for Russian (Kutuzov et al., 2021), Italian (Basile et al., 2020), Norwegian (Kutuzov et al., 2022a) and Spanish (Zamora-Reina et al., 2022), where we see stronger indications that contextualized models perform better than type-based ones.

2.2 Contextualized models training

Training contextualized models requires a massive amount of textual data, prolonged training time, and considerable computational power. All these have made the training of new models a complicated procedure available only to selected research labs over the world, as oftentimes researchers lack the necessary resources to train their own models (our interest in historical language poses a particular challenge in this regard, as historical texts are usually much smaller in size).

To mitigate these requirements, speed up the training process, and increase the usability of these models, *fine-tuning* was developed. Using fine-tuning, models that were already trained (now called pre-trained models) are continued to be trained, albeit on much smaller data and sometimes on a different task (pre-trained models are usually trained on standard masked-word and next-sentence prediction tasks).

This two-step training setup was found to greatly improve the state-of-the-art performance in many tasks and today, the use of fine-tuning in contextualized models has become the dominant paradigm in NLP (Howard and Ruder, 2018; Devlin et al., 2019; Merchant et al., 2020). Importantly, it was found that fine-tuning, henceforth FT, can *transfer* to other tasks and languages and thus improve performance on tasks and datasets it was not trained on (Peters et al., 2019; Khashabi et al., 2020), presumably because of shared information that is required to process these different tasks. In this paper, we aim to leverage the trans-

[2] https://github.com/ChangeIsKey/LSC-AGG

fer capabilities and test whether FT a contextualized model on a range of NLP tasks improves its performance to detect LSC.

3 Method

Our aim is to test whether a state-of-the-art method for detecting lexical semantic change, based on pre-trained contextual embeddings, can be improved by adding fine-tuned layers. Therefore, we start with a BERT model and detect semantic change following Kutuzov and Giulianelli (2020). We chose their model as it was the best-performing system in English in the post-evaluation stage of SemEval-2020 Task 1. Next, we add information from fine-tuning for tasks that go beyond a masked language model objective. We include tasks aimed to capture both low-level linguistic information like part-of-speech tagging, as well as higher-level (semantic) information such as sentiment analysis, linguistic inference, and machine reading comprehension.

3.1 LSC method

The task of detecting lexical semantic change can be described as the following: given two corpora C_1 and C_2 from time periods T_1 and T_2, as well as a set of target words, detect which words have changed between T_1 and T_2 as evidenced in C_1 and C_2. This is a special case of the general LSC problem which includes arbitrarily many time periods $T_1, \ldots, T_N$.

Following Kutuzov and Giulianelli (2020), we use a pre-trained BERT base model to generate the contextualized embeddings of each occurrence of the target words in C_1 in C_2, resulting in two corresponding embedding matrices U_w^{t1} and U_w^{t2}. Given these embedding matrices, we calculate the change scores of each target word in one of two ways: inverted cosine similarity over word prototypes (PRT); and average pairwise cosine distance between token embeddings (APD).

$$PRT(U_w^{t1}, U_w^{t2}) = \frac{1}{d(\frac{\sum_{x_i \in U_w^{t1}} x_i}{N_w^{t1}}, \frac{\sum_{x_j \in U_w^{t2}} x_j}{N_w^{t2}})} \quad (1)$$

$$APD(U_w^{t1}, U_w^{t2}) = \frac{1}{N_w^{t1} N_w^{t2}} \sum_{x_i \in U_w^{t1}, x_j \in U_w^{t2}} d(x_i, x_j) \quad (2)$$

N_w^{t1}, N_w^{t2} stands for the number of occurrences of w in $T1$ and $T2$. d is the cosine distance. For both methods, higher values suggest a larger semantic change.

3.2 Fine-tuning

The main contribution of this paper is the injection of richer meanings into contextualized embeddings using fine-tuning. Our fine-tuned models are derived mostly from adapters (Pfeiffer et al., 2020; Poth et al., 2021), which are trained layers that can be integrated directly into transformer-based models (the most popular type of contextualized models) in order to perform different tasks. Using adapters enabled us to speed up our experiments as they are readily available[3] and can be seamlessly integrated into the tested models.

In addition to using adapters, we also fine-tune two models locally on sentiment classification and part-of-speech tagging in order to compare the performance of fine-tuned models with adapter-based models. For sentiment classification, we use the sst2 dataset (Pang and Lee, 2004) while for part-of-speech tagging we use CoNLL2003 (Tjong Kim Sang and De Meulder, 2003). Since there is no test set with gold labels for sst2, we randomly sample 30% of the data from the validation set as a test set. The accuracy of the fine-tuned models on the test set is 0.908 for sentiment analysis and 0.931 for part-of-speech tagging. We use the BERT-base-uncased model for all our experiments, both with adaptors and local FT. Table 1 details the FT tasks we used.

Task/Model	Type
natural language inference (nli)	pf
machine reading comprehension (reading compre)	pf
sentiment (sst2)	pf
sentiment (sst2-pfeiffer)	pfeiffer
sentiment (sst2-hously)	hously
semantic textual similarity	pf
linguistic acceptability	pf
grammatical error correction (error detect)	pf
semantic tagging	pf
named entity recognition (ner)	pf
part-of-speech tagging (pos)	pf
phrase chunking	pf
sentiment (sst2-fine-tune)	fine-tuned
part-of-speech tagging (pos-fine-tune)	fine-tuned

Table 1: Fine-tuned models & tasks for adapters (upper) and locally trained FT (lower). Type refers to adaptors trained by Poth et al. (2021)(pf), Peiffer and Hously. Task abbreviations in parentheses.

[3] https://adapterhub.ml/

4 Evaluation

For our experiments, we use a standard evaluation dataset for LSC.

4.1 Evaluation data

We use the English dataset of SemEval-2020 Task 1 (Schlechtweg et al., 2020) for unsupervised lexical semantic change detection. The task was the first of its kind to provide manually annotated gold data for the purpose of fair and comparable evaluation of methods for LSC. The task consists of two sub-tasks aimed to measure change between two time-specific corpora C_1 and C_2:

Binary Classification: for a set of target words, decide which words lost or gained sense(s) between C_1 and C_2, and which did not.

Ranking: rank a set of target words according to their degree of LSC between C_1 and C_2.

These tasks are related but complementary. The ranking task measures the degree of change and takes into consideration lost or gained senses, but also includes changes in existing senses (e.g., by means of broadening or narrowing) which the binary classification task does not consider.

The English dataset of SemEval consists of 37 target words derived from the Clean Corpus of Historical American English (CCOHA) (Davies, 2012; Alatrash et al., 2020). The two 50-year periods are $C_1 = 1810{-}1860$ and $C_2 = 1960{-}2010$ from which each target word has a set of 100 randomly sampled sentences. These sentences have been compared by human annotators and ranked on a scale for lexical semantic change. Based on the outcome of the roughly 29,000 human judgments, words are classified as changing or stable and assigned a change degree. The process is described in detail by Schlechtweg et al. (2021)

4.2 Evaluation Metrics

We use two evaluation metrics.

Spearman correlation is used to compute the rank correlation between model predictions and gold labels in the ranking task.

AUC & ROC are used to evaluate the impact of different thresholds on the model performance in the binary classification task.

4.3 Validation through permutation

The work presented in this paper suggests that certain combinations of FT tasks improve LSC detection for both ranking and classification tasks. Importantly, these combinations are chosen based on improved performance in the SemEval tasks. Ideally, we would consider this as the training set and then test the chosen combinations on a held-out dataset to examine if similar gains are acquired. However, such a test set is lacking, and cannot be constructed via standard train-test splits from the 37 words, from which only 16 have changed.[4] We propose permutation tests to mitigate this shortcoming, and to enable us to draw reliable conclusions from our study despite this limitation.

For both tasks, we evaluate the probability that the best combinations we report were found by chance. We conduct a permutation analysis and generate artificial FT task scores that are based on the distributions of the existing FT values from the 14 FT models (Table 1). We then compute the relevant evaluation metric (Spearman or AUC) for each artificial FT, and repeat the process 100,000 times, creating a distribution of ranking or classification performance scores. We then compute the proportion of times that the artificial random FT combinations performed better than our best combinations, in the form of a p-value for our chosen combinations. Ultimately, this allows us to test the statistical significance of our results, and evaluate how likely it is that our best combinations were found by chance.

5 Experiments

Two models are used as baselines, relative to which we test if adding FT (either adaptors or local FT) improves performance. One of the baseline models was used as the basis on top of which the different FT combinations were tested.

We choose the best combination of FT tasks by analyzing the results of the *ranking task* and then test it on the *binary classification task* using a modified decision criterion (i.e., ROC analysis). Because the two tasks are different, this allows us to use the latter as an ad-hoc evaluation test.

[4] Under these conditions it was also not feasible to conduct a systematic regression analysis, which would have lacked the statistical rigor to reach reliable results.

5.1 Baselines

We make use of two rather different baseline models that are used together with the fine-tuning. Kutuzov and Giulianelli (2020) whose model scored the highest in the English part of SemEval (henceforth BERT), and HistBERT (Qiu and Xu, 2022) which provides a contextualized model pre-trained on historical English.[5] Together, they provide complementing baselines to test our research hypothesis. All FT combinations where made on top of the BERT baseline.

We also compute p-values from the permutation tests for the two tasks (see Section 4.3), and for each method (APD and PRT).

5.2 Ranking task

Given a fine-tuning task FT_i, we obtain two embedding matrices U_w^{t1} and U_w^{t2} for each target word. We use these embedding matrices to calculate the semantic change score of a target word by means of APD and PRT. Once we have the change scores for all target words, we produce a ranking of the words. We then measure the performance as Spearman correlation of the change score ranks compared to the gold ranking. This is illustrated in the following formula, where P_{ind} stands for the performance of the individual task. $Score$ is the scoring function (AUC or Spearman correlation).

$$P_{ind} = Score(FT_i, gold) \tag{3}$$

There are in total $\text{FT}_1, \ldots, \text{FT}_{14}$ fine-tuning tasks. In addition to their individual performance, we are interested if they add complementary information, and therefore want to measure their combined performance. We thus enumerate all the possible combinations of the tasks. For each combination, we average the change scores produced by each participating FT_i. For instance, we can combine change scores derived from Natural Language Inference and Named Entity Recognition by averaging the scores for each target. There are in total FT_{14}^1 + FT_{14}^2 + $\ldots$ + FT_{14}^{13} + FT_{14}^{14} combinations. We measured the performance of each combination by means of its Spearman correlation. This is illustrated in the following formula, where c is the combination of different tasks.

$$P_c = Score\left(\frac{1}{|c|}\sum_{i \in c}(FT_i), gold\right) \tag{4}$$

[5]There are four versions: HistBERT-prototype, HistBERT-5, HistBERT-10, and HistBERT-full. They differ in the size and time period of the training data. In this study, we report the averaged scores of the four models.

Method	Rank-PRT	Rank-APD
ner	0.218	0.285
nli	**0.427****	0.634
pos	0.205	0.205
error detect	0.352	0.593
linguistic acceptability	0.364	0.622
phrase chunking	0.076	0.185
pos-fine-tune	0.277	0.087
reading compre	0.416	0.636
semantic tagging	0.265	0.255
sst2	0.422	0.608
sst2-hously	**0.435****	0.627
sst2-fine-tune	0.123	0.210
sst2-pfeiffer	0.391	0.459
textual similarity	0.378	0.694
BERT	0.423	**0.706**
HistBERT-ave	0.264	0.441

Table 2: Spearman correlations for different FTs on the ranking tasks. Best FTs in bold. Statistical significance marks *, **, ***: for p-values<.05, .01, .001, respectively.

5.3 Binary Classification task

For the binary classification task, we calculate the AUC score of each FT_i. One advantage of the AUC score over accuracy is that we do not need to define the threshold to determine a word changes or not, given the change scores derived from PRT and APD are continuous values

We carry out two experiments here: 1) testing the best models found in the ranking task on the binary classification task, and 2) examining the performance of individual FT_i as well as combined models. Our motivation for the first experiment is that we want to evaluate our best models from a new perspective given that the ranking and classification tasks feature different task profiles. In the second experiment, we focus on the combination effect, and take the more challenging case, examining the FTs with the highest as individual FT tasks.

6 Results

6.1 Ranking task results

We begin by reporting the results of individual FT_i. The results are presented in Table 2. For PRT, the range of correlation between (the ranking produced by) each FT_i and the gold rank is

Method	Combination(s)	Correlation
APD best 5	BERT, nli, textual similarity, error detect , pos-fine-tune	0.723***
	BERT, sst2, error detect, pos-fine-tune	0.722***
	sst2, textual similarity, error detect	0.722***
	BERT, sst2, textual similarity, error detect	0.721***
	BERT, textual similarity, error detect	0.721***
APD baseline	BERT	0.706
	BERT, random scores	0.462
	HistBERT (averaged)	0.441
PRT best 5	nli, pos-fine-tune, sst2-fine-tune	0.531**
	nli, sst2-pfeiffer, pos-fine-tune, ner, sst2-fine-tune	0.515*
	BERT, nli, sst2-pfeiffer, ner, sst2-fine-tune	0.503*
	nli, sst2-pfeiffer, pos-fine-tune, sst2-fine-tune	0.503*
	nli, reading compre, sst2-pfeiffer, histBERT-10, pos-fine-tune, sst2-fine-tune	0.502*
PRT baseline	BERT	0.423
	BERT, random scores	0.336
	HistBERT (averaged)	0.264

Table 3: Ranking results for 5 best FT combinations, APD and PRT. p-values as reported in Table 2.

0.427 – 0.076. We find that most fine-tuned models do not beat the BERT baseline, with only 2 exceptions (nli and sst2-hously). For APD, individual FT$_i$s range from 0.694 to 0.086, with one FT having a negative correlation of 0.211. Here, the maximum baseline (BERT) is marginally higher than any individual FT with a correlation value of 0.706. Overall, the results from Table 2 show that most individual FTs do not improve task performance further for both PRT and APD.

We now turn to combining different FT for the ranking task, shown in Table 3. For PRT, the five best models (0.531 – 0.502) all rank higher than the baseline models (0.423 – 0.264). For APD, the five best models (0.723 – 0.721) also rank higher than the baseline models (0.706 – 0.441).

We note that although the performance gains are statistically significant for both PRT and APD, they are much more prominent for PRT. We also note that not every combination leads to an improvement. Some tasks (or task combinations) can yield lower performance. For instance, combing part-of-speech and sentiment in APD actually deteriorates task performance. More details can be found in the appendix.

The random permutations, where we average BERT with scores sampled from the overall score distribution, over 100,000 runs, corroborate our findings. For PRT we get a mean correlation of 0.336 (s.d. of 0.107), and less than 1 per 100 randomly sampled scores perform better than the best PRT (0.531) (p-value$<$0.01). For APD, the corresponding values are 0.462 (s.d. of 0.147), and less than 2 per 1000 randomly sampled scores perform better (p-value$<$0.01).

6.2 Classification task results

Method	AUC-PRT	AUC-APD
ner	**0.688***	0.604
nli	0.646	0.714
pos	0.634	0.536
error detect	0.634	0.670
linguistic acceptability	0.628	0.676
phrase chunking	0.622	0.622
pos-fine-tune	0.634	0.643
reading compre	0.670	0.696
semantic tagging	0.649	0.631
sst2	0.658	0.664
sst2-hously	**0.682***	0.685
sst2-fine-tune	**0.673***	0.417
sst2-pfeiffer	0.631	0.613
textual similarity	0.661	**0.741****
BERT	0.673	0.717
HistBERT-ave	0.657	0.659

Table 4: AUC scores for different FTs on the classification task. Best FTs in bold. p-values as reported in Table 2.

We begin by reporting the results of individual FT$_i$, shown in Table 4. We report the AUC scores to avoid threshold selection. We observe more variance in APD than in PRT in terms of model performance. For PRT, we observe that named entity recognition provides the highest AUC score (0.688) while phrase chunking generates the lowest performance (0.622). As for APD, textual similarity was the only FT to surpass the baseline and achieve statistical significance with AUC score of

Method	Combination(s)	AUC
APD best 3	nli, textual similarity	0.741*
	textual similarity, nli, BERT	0.738*
	textual similarity, BERT	0.735*
APD ind best	textual similarity	0.741*
PRT best 3	ner sst2-hously, sst2-fine-tune *(p-value = 0.054)*	0.732
	ner, sst2-fine-tune, reading compre *(p-value = 0.090)*	0.720
	ner, sst2-hously, sst2-fine-tune, reading compre *(p-value = 0.161)*	0.714
PRT ind best	ner *(p-value = 0.161)*	0.688

Table 5: Results of combining different FTs on the classification task. We present the best individual task performance (APD/PRT individual best) for comparisons. * means statistically significant improvement over the BERT baseline.

0.741. Similar to what we found for the ranking task, it seems that most individual FTs do not improve task performance neither for PRT or APD.

To test the performance of combinations, we take the top 5 best-performing models for PRT and APD separately. We then enumerate all possible combinations from them and report the results in Table 5. We find that combining different FTs improves task performance in PRT but not in APD. In PRT, we see a five percentage of AUC increase when we combine Name Entity Recognition, sst2-hously and sst2-fine-tune (from 0.688 to 0.732). In APD, we do not observe performance gains over the individual FTs.

6.3 Transferability scenario

As the ranking and classification tasks are curated differently (see Section 4.1), we see this as an opportunity to use the latter as an ad-hoc evaluation set by testing the transferability between the two tasks. We ask: are the best combinations that we found for ranking also useful (i.e., can be transferred) for the purpose of classification? We apply the three best combinations found in Section 6.1 to the classification task, plot their ROC curves and calculate their AUC scores. The results are shown in Figure 1. We find that the best models found in the ranking task outperform BERT baseline in the classification task. In PRT, the best model achieves an AUC score of 0.774 compared with the base model (0.673). In APD, though less obvious, the best model still performs better than the baseline (0.744 v.s. 0.717). While the ranking and classification tasks are designed for different purposes, this experiment suggests that the analysis of ranking results can guide the choice, and thus transfer, of models for the classification task.

7 Discussion and Conclusion

In this paper, we investigate our hypothesis that adding linguistic information to pre-trained language models by means of fine-tuning can lead to improved performance on unsupervised Lexical Semantic Change detection. We chose two classic LSC tasks, ranking and binary classification, from SemEval-2020 Task 1 (Schlechtweg et al., 2020). To simplify and speed up the process of fine-tuning we used adaptors (Pfeiffer et al., 2020), which are pre-trained fine-tuning modules that can augment existing contextualized models and are readily and freely available for English.

First, we tried single FT tasks, which showed little or no improvement over the BERT baseline. Then we combined several FT tasks together by means of simple averaging and found considerable improvements.

Adding linguistic information, like part-of-speech, to a standard (masked) language model can offer additional information that improves the ability of the models to detect lexical semantic change. However, some kinds of information are adjacent to semantic change and therefore make the models capture change, but not necessarily semantic change. In future work, we will conduct in-depth analysis of the worst models, to see what information they capture instead and why this information seems to hurt the performance of the LSC model.

Our work is not the first to introduce linguistically augmented contextualized models for the task of LSC. Giulianelli et al. (2022) used an ensemble method to inject linguistic information, and reported performance gains in LSC tasks. However, they focus on "low-level" morpho-syntactic information. Our approach, in addi-

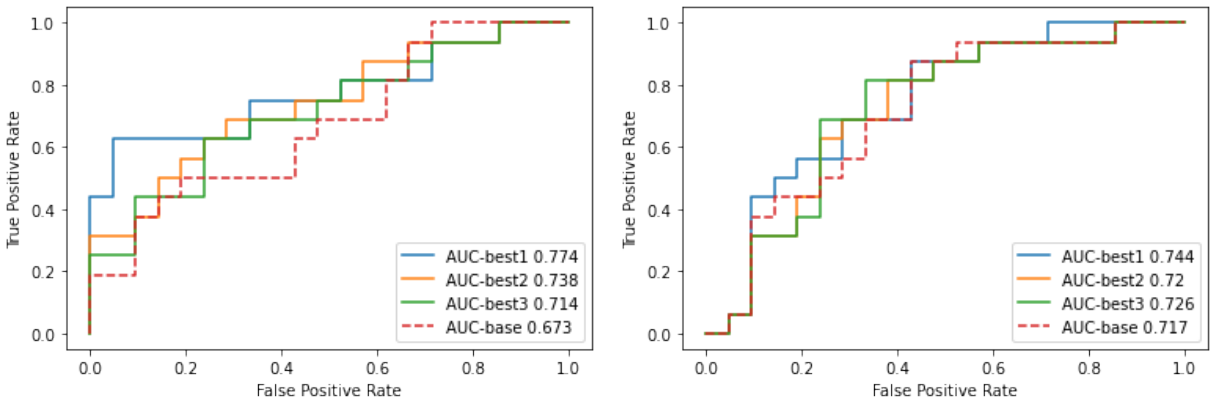


Figure 1: ROC curves of the best 3 models in the ranking task as they perform on the classification task. Left: PRT, right: APD. See Table 3 for combinations details.

tion to using a completely different ML method for linguistic augmentation, spans both ends of the linguistic-informative spectrum, ranging from part-of-speech, to sentiment to logical inference.

One deficiency of our results is that they are based on a small evaluation dataset, which means that the improvements we report could be attributed to chance (or model over-fitting). To mitigate this concern, and add scientific rigour to our analyses, all results were tested in permutation tests and are reported with their p-values. Evaluation is also done with comparison to two strong baseline models, each of which provides different perspective to test our research hypothesis. Outperforming the Kutuzov and Giulianelli (2020) model suggests that enriching models with additional linguistic information is highly beneficial, and outperforming HistBERT supports the idea that this information cannot be gained even when a model is exclusively (pre-) trained on historical text. Overall our results clearly suggest that LSC models can be improved dramatically with relatively simple steps of fine-tuning on a range of standard linguistic tasks.

We note that there are differences between the best performing FTs for the PRT and APD methods. Although certain FTs are shared (e.g., NLI and POS), others appear more systematically in PRT or APD. We interpret these inconsistencies as stemming from differences between the PRT and APD methods themselves, and do not view them as negative. Instead, each method enriches the baseline with different types of information and hence allows the model to capture slightly different aspects of LSC. Combined with the FTs, the final results can be quite different. This complementing view of the two LSC methods is supported by the results of Kutuzov et al. (2022b), who recently reported that joining PRT and APD improves LSC detection results. The most probable explanation for this is that the two LSC methods are sensitive to different aspects of change.

From a theoretical point of view, our conclusions are inline with how linguists describe the phenomenon of LSC. Linguistic theory distinguishes between different types of LSC, and emphasizes that changes are never "general" but pertinent to certain aspects of meaning. Therefore, computationally analyzing words' meaning change using a "single ruler" as is done by current state-of-the-art LSC models, may be insufficient to describe the richness and diversity of change. We believe our findings provide an inroad for extending the capacity of LSC models and encouraging future research in this direction.

This is but the first step in exploring the potential of using FT to enrich and improve contextualized models of LSC. In our future work we will corroborate these findings more rigorously: extending these to other languages, testing the generalizability of the chosen FT tasks across LSC models, and test our approach in the discovery of new cases of words that change their meaning to go beyond a small set of examples.

Acknowledgement

This work was supported in part by the Riksbankens Jubileumsfond (under reference number M21-0021, *Change is Key!* program), and in part by the Swedish Research Council (2019–2022; contract 2018-01184, *Towards Computational Lexical Semantic Change Detection* project).

References

Reem Alatrash, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2020. CCOHA: Clean Corpus of Historical American English. In *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC'20)*. European Language Resources Association (ELRA).

Rabab Alkhalifa, Elena Kochkina, and Arkaitz Zubiaga. 2023. Building for tomorrow: Assessing the temporal persistence of text classifiers. *Information Processing & Management*, 60(2):103200.

Nikolay Arefyev and Vasily Zhikov. 2020. BOS at SemEval-2020 Task 1: Word Sense Induction via Lexical Substitution for Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Francesco Barbieri, Jose Camacho-Collados, Bhuwan Dhingra, Luis Espinosa-Anke, Elena Gribovskaya, Angeliki Lazaridou, Daniel Loureiro, and Leonardo Neves, editors. 2022. https://aclanthology.org/2022.evonlp-1.0 *Proceedings of the The First Workshop on Ever Evolving NLP (EvoNLP)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid).

Pierpaolo Basile, Annalina Caputo, Tommaso Caselli, Pierluigi Cassotti, and Rossella Varvara. 2020. Diacr-ita@ evalita2020: Overview of the evalita2020 diachronic lexical semantics (diacr-ita) task.

Mark Davies. 2012. Expanding Horizons in Historical Linguistics with the 400-Million Word Corpus of Historical American English. *Corpora*, 7(2):121–157.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. https://doi.org/10.18653/v1/N19-1423 BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Haim Dubossarsky, Daphna Weinshall, and Eitan Grossman. 2016. Verbs change more than nouns: A bottom up computational approach to semantic change. *Lingue e Linguaggio*, 1:5–25.

Mario Giulianelli, Andrey Kutuzov, and Lidia Pivovarova. 2022. https://doi.org/10.18653/v1/2022.lchange-1.6 Do not fire the linguist: Grammatical profiles help language models detect semantic change. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 54–67, Dublin, Ireland. Association for Computational Linguistics.

Amaru Cuba Gyllensten, Evangelia Gogoulou, Ariel Ekgren, and Magnus Sahlgren. 2020. SenseCluster at SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. https://doi.org/10.18653/v1/P18-1031 Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Daphna Keidar, Andreas Opedal, Zhijing Jin, and Mrinmaya Sachan. 2022. Slangvolution: A causal analysis of semantic change and frequency dynamics in slang. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1422–1442. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. https://doi.org/10.18653/v1/2020.findings-emnlp.171 UNIFIEDQA: Crossing Format Boundaries with a Single QA System. In *ACL*, pages 1896–1907.

Andrey Kutuzov and Mario Giulianelli. 2020. https://doi.org/10.18653/v1/2020.semeval-1.14 UiO-UvA at SemEval-2020 task 1: Contextualised embeddings for lexical semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 126–134, Barcelona (online). International Committee for Computational Linguistics.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: A survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Andrey Kutuzov, Lidia Pivovarova, et al. 2021. Rushifteval: a shared task on semantic shift detection for russian. In *Computational linguistics and intellectual technologies: Papers from the annual conference Dialogue*. Redkollegija sbornika.

Andrey Kutuzov, Samia Touileb, Petter Mæhlum, Tita Enstad, and Alexandra Wittemann. 2022a. https://aclanthology.org/2022.lrec-1.274 NorDiaChange: Diachronic semantic change dataset

for Norwegian. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2563–2572, Marseille, France. European Language Resources Association.

Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. 2022b. https://doi.org/10.3384/nejlt.2000-1533.2022.3478 Contextualized embeddings for semantic change detection: Lessons learned. *Northern European Journal of Language Technology*, 8(1).

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. https://doi.org/10.18653/v1/2020.blackboxnlp-1.4 What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2004. https://doi.org/10.3115/1218955.1218990 A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. https://doi.org/10.18653/v1/W19-4302 To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. https://www.aclweb.org/anthology/2020.emnlp-demos.7 Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Clifton Poth, Jonas Pfeiffer, Andreas R"uckl'e, and Iryna Gurevych. 2021. https://aclanthology.org/2021.emnlp-main.827 What to pre-train on? Efficient intermediate task selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10585–10605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wen Qiu and Yang Xu. 2022. Histbert: A pretrained language model for diachronic lexical semantic analysis. *ArXiv*, abs/2202.03612.

Guy Rosin and Kira Radinsky. 2022. Temporal attention for language models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1498–1508.

Dominik Schlechtweg, Stefanie Eckmann, Enrico Santus, Sabine Schulte im Walde, and Daniel Hole. 2017. German in flux: Detecting metaphoric change via word entropy. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pages 354–367, Vancouver, Canada.

Dominik Schlechtweg, Barbara Mcgillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. Semeval-2020 task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the international Conference on Computational Linguistics (COLING)*.

Dominik Schlechtweg, Nina Tahmasebi, Simon Hengchen, Haim Dubossarsky, and Barbara McGillivray. 2021. https://doi.org/10.18653/v1/2021.emnlp-main.567 DWUG: A large resource of diachronic word usage graphs in four languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7079–7091, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2021. https://doi.org/10.5281/zenodo.5040302 Survey of computational approaches to lexical semantic change detection. In Nina Tahmasebi, Lars Borin, Adam Jatowt, Yang Xu, and Simon Hengchen, editors, *Computational approaches to semantic change*. Language Science Press.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. https://aclanthology.org/W03-0419 Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Frank D. Zamora-Reina and Felipe Bravo-Marquez. 2020. DCC-Uchile at SemEval-2020 Task 1: Temporal Referencing Word Embeddings. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Frank D. Zamora-Reina, Felipe Bravo-Marquez, and Dominik Schlechtweg. 2022. https://aclanthology.org/2022.lchange-1.16/ Lscdiscovery: A shared task on semantic change discovery and detection in spanish. In *Proceedings of the 3rd International Workshop on Computational Approaches to Historical Language Change*, Dublin, Ireland.

A Appendix

Method	Combination(s)	Correlation
APD worst 3	pos-fine-tune, sst2-fine-tune	-0.214
	sst2-fine-tune	-0.210
	pos-fine-tune	0.087
APD baseline	BERT	0.706
	BERT, random scores	0.462
	HistBERT (averaged)	0.441
PRT worst 3	sst2-fine-tune, histBERT (full)	0.012
	pos-fine-tune, sst2-fine-tune, histBERT (full)	0.014
	pos-fine-tune, histBERT (full)	0.038
PRT baseline	BERT	0.423
	BERT, random scores	0.336
	HistBERT (averaged)	0.264

Table 6: Ranking results for 3 worst FT combinations

Question Answering and Question Generation for Finnish

Ilmari Kylliäinen and **Roman Yangarber**
University of Helsinki, Finland
Department of Digital Humanities
first.last@helsinki.fi

Abstract

Recent advances in the field of language modeling have improved the state-of-the-art in question answering (QA) and question generation (QG). However, the development of modern neural models, their benchmarks, and datasets for training them has mainly focused on English. Finnish, like many other languages, faces a shortage of large QA/QG model training resources, which has prevented experimenting with state-of-the-art QA/QG fine-tuning methods. We present the first neural QA and QG models that work with Finnish. To train the models, we automatically translate the SQuAD dataset and then use normalization methods to reduce the amount of problematic data created during the translation. Using the synthetic data, together with the Finnish partition of the TyDi-QA dataset, we fine-tune several transformer-based models to both QA and QG and evaluate their performance. To the best of our knowledge, the resulting dataset is the first large-scale QA/QG resource for Finnish. This paper also sets the initial benchmarks for Finnish-language QA and QG.

1 Introduction

The purpose of question answering (QA) systems is to help users find information more efficiently. QA systems come in many forms and offer help in everything from database querying to complex information search from the entire World Wide Web. Recently, much attention has been directed toward developing extractive QA models that can draw answers directly from spans of text. Popular approaches have emerged that integrate components that first retrieve documents relevant to a question, with models for reading comprehension that pinpoint the answers in the retrieved documents.

A task closely related to QA, yet less researched, is question generation (QG), where the object is to generate natural and grammatical questions that can be answered by a specific answer using some given context. QG can be used to, e.g., automatically create reading comprehension tasks, or to improve the interactivity of virtual assistants. It can also be used as a data augmentation tool—to create new training data for QA systems.

Recently, the focus for both tasks has moved to neural language models utilizing transfer learning—e.g., BERT (Devlin et al., 2019) or XLNet (Yang et al., 2019), at least for languages such as English. Despite the advances in QA and QG, the lack of training datasets has hindered the use of state-of-the-art deep learning methods to develop modern QA and QG models for Finnish. Finnish, like many languages, lacks the resources to train models for the two tasks. In fact, no monolingual Finnish QA or QG models have been reported to exist at all.

In order to fine-tune models for Finnish extractive QA and answer-aware QG, we first create a Finnish QA dataset by automatically translating the SQuAD—**S**tanford **Q**uestion **A**nswering **D**ataset dataset (Rajpurkar et al., 2016), from English to Finnish, and then use automatic normalization to clean up problematic data. We use the synthetic data to train several transformer-based models for QA and QG and evaluate their performance. We release the data to the research community to support future research.[1]

The paper is organized as follows: in Section (2) we review prior work on QA, QG, and generation of synthetic resources. In Section 3, we review the dataset creation, and introduce additional datasets used to train and evaluate the models. Section 4 reviews the fine-tuning methods, and Section 5 discusses the results of the experiments. Section 6 concludes and offers directions for future work.

[1] https://huggingface.co/datasets/ilmariky/SQuAD_v2_fi

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 529–540

2 Related Work

2.1 QA and QG for Other Languages

Approaches to both question answering and question generation have significantly evolved throughout their history. More recently, along with new datasets and novel deep learning methods, neural approaches have become the state of the art for both tasks.

It has become popular for information retrieval-based QA systems to incorporate a neural machine reading comprehension (MRC) component that extracts answers from a set of retrieved documents. After the introduction of the transformer architecture, models like BERT (Devlin et al., 2019) have become a popular tool for the answer extraction task. Many models have already surpassed human performance on the SQuAD1.1 dataset (Yamada et al., 2020; Yang et al., 2019) and some models can also predict whether the passage contains the answer to the question at all (Zhang et al., 2020). Lee et al. (2019) presented a unified end-to-end architecture capable of both retrieving and reading.

Since the mid-2010s, many RNN-based approaches have been proposed to QG (Zhou et al., 2017; Du et al., 2017; Zhao et al., 2018). However, the Transformer architecture (Vaswani et al., 2017) solved many problems that RNNs have, and has also become a popular architecture for QG models. The QG system by Wang et al. (2020) employs the encoder and the decoder from the Transformer. They combine the question generation and answer selection process in a joint model and treat the answers as a hidden pivot for question generation. Durmus et al. (2020) fine-tune a pre-trained BART model (Lewis et al., 2020) to generate questions from sentences. Chan and Fan (2019b) fine-tune a BERT model to work in a sequential manner to generate questions from paragraphs of text. Their model achieved state-of-the-art results in paragraph-level QG.

2.2 QA and QG for Finnish

Very little research on Finnish QA exists to date. Aunimo et al. (2004) presented two cross-lingual QA systems, `Tikka` and `Varis`, that took Finnish questions as input and found answers to them from a collection of English-language documents. `Tikka` is a simple baseline model, while `Varis` is more sophisticated. The pipelines of both systems start with defining the question type with the use of syntactic information and then translating the question into English. `Varis` also tries to extract the answer type of the question using a named entity recognizer. `Tikka` and `Varis` could correctly answer 22.5% and 29.0% of the questions presented to them, respectively.

No previous work is found on monolingual or cross-lingual QG systems that work with Finnish. Therefore, to the best of our knowledge, the results reported in this paper are the first ones for Finnish-language question generation.

2.3 Generation of Synthetic QA Corpora

Large annotated corpora are essential for fine-tuning pre-trained deep architecture but, unfortunately, they are also scarce for Finnish. In the context of QA, generation of synthetic corpora often means creation of a dataset via, e.g., automatic or semiautomatic translation of an existing QA dataset, or automatic data extraction from raw unlabeled data.

Recently, there have been several attempts to create synthetic datasets for QA. Carrino et al. (2020) translated an English QA dataset automatically to Spanish using a method called Translate-Align-Retrieve. The method is based on MT and an unsupervised alignment algorithm. Alberti et al. (2019) combined QG and answer extraction models with a technique they refer to as roundtrip consistency-ensuring filtering to automatically create a synthetic English QA dataset from unlabeled text passages. Abadani et al. (2021) translated the SQuAD2.0 QA dataset (Rajpurkar et al., 2018) automatically into Persian, and then finalized the data into two datasets, of which one is corrected manually and the other automatically. The automatically corrected one is many times bigger and also yielded better results. The SQuAD dataset has also been automatically translated to Swedish (Okazawa, 2021) and French (Kabbadj, 2018).

3 Data

3.1 SQuAD

SQuAD is a large English QA dataset created for training machine learning models for the extractive QA task. It is one of the most popular QA datasets, and many other QA datasets have followed its methodology (Clark et al., 2020; d'Hoffschmidt et al., 2020; Lim et al., 2019). SQuAD has also been a popular resource for answer-aware neural question generation (NQG) (Chan and Fan, 2019a; Du et al., 2017; Klein and Nabi, 2019).

	English	Finnish translation
Passage	The capital, Brazzaville, is located on the Congo River, in the south of the country, immediately across from Kinshasa, the capital of the Democratic Republic of the Congo.	Pääkaupunki Brazzaville sijaitsee Kongo-joen varrella maan eteläosassa, vastapäätä Kongon demokraattisen tasavallan pääkaupunkia Kinshasaa.
Question	What country does Kinshasa serve as capital of?	Minkä maan pääkaupunki Kinshasa on?
Answer	Democratic Republic of the Congo	Kongon demokraattinen tasavalta

Table 1: An example of problematic data resulting from translating passages and answers separately. The translated answer (in the nominative case) is not found within the translated passage (where it appears in the genitive case) which is required for extractive QA.

The first version of SQuAD (SQuAD1.1) contains over 100K passage-question-answer triplets that crowdworkers extracted from 536 Wikipedia articles. Each article is divided into several passages, and each passage has several questions related to its contents. Each question is linked with an answer (a substring of the passage) and the position of the answer's first character in the passage. The second version of the dataset, SQuAD2.0, contains additional 50K questions, similar to the first version's questions but impossible to answer with the given passage. The extension's idea was to enable the development of models that can identify unanswerable questions.

3.2 Dataset Translation and Normalization

We translated all the text data in the SQuAD2.0 into Finnish using Google NMT (Wu et al., 2016) with the Google Translate API. The passage, questions, and answers were translated separately, which led to many of the translated answers not being substrings of the translated passage. That was sometimes caused by translation errors, but one major factor was that the data was translated from a weakly inflected, analytic language to a highly inflected, agglutinative language. In other words, the MT system has no way of knowing how to inflect the words in the translation without any context. The SQuAD format requires the answer to be a substring of the passage as it is an extractive QA dataset. The problem is illustrated in Table 1. Okazawa (2021) used a simple highlighting technique to tackle this problem when translating SQuAD2.0 into Swedish. Rather than translating the passage and the answer separately, they put special markers (`[0]`) around the answer substring before the translation and afterward simply extracted the translated answer span between the markers and then removed the markers. However, using it would have required translating the same passages multiple times with different answers marked since passages are linked with several questions. This was not feasible solely because using Google NMT via API is not free.

After translation, we used simple normalization methods to identify the answer substring in the translated passage whenever it did not contain the separately translated answer. In total, there were four normalization steps: regular expressions, lemmatization, stemming, and using the English answer.. The script started with the first one and moved to the next one if necessary.

In the first step, a set of regular expressions was used to fix some inconsistencies (in, e.g., white spaces and punctuation) that were found to occasionally occur in the translations. In the next step, both the passage and the answer were lemmatized, and the script checked whether the now lemmatized answer was included in the lemmatized passage. If lemmatization did not lead to a match, the script moved to the next step: stemming. Stemming was done because the lemmatizer was observed to not recognize many of the passage words as they were often proper nouns. If no match was found after stemming, the last step was to check whether the English answer was included in the translated passage; if it was, it was used as the answer with the assumption that the English answer was mistakenly translated. This was often the case with, e.g., English song and movie names when they were translated with no context. If no match was found after all normalization, the question-answer pair was discarded from the final dataset.

If there was a match at any normalization step, the script proceeded to search its location in the passage. The answer search started from the English answer's relative position in the translated passage and continued to neighboring positions un-

til the answer was found. This was done to reduce the chance of choosing the starting position of a wrong occurrence, as some passages contain the answer string multiple times in different positions. After finding the answer start position, the question-answer pair was added to the final dataset.

With the normalization procedure, roughly 32K answers were modified to match the passage strings. The data consists of 101,120 passage-question-answer triplets that are valid in the sense that the answers are included in the passages. 66K of them are answerable (from SQuAD1.1), and 34K are unanswerable with the given passage (from SQuAD2.0). This means that roughly 28% of the data included in the publicly available partition of SQuAD1.1 (92K questions) had to be discarded. The amount is approximately the same when taking into account also the "unanswerable" questions of SQuAD2.0.

3.3 Finnish TyDi-QA Corpus

TyDi-QA—**T**ypologically **D**iverse **Q**uestion **A**nswering (Clark et al., 2020), consists of two QA datasets, covering 11 typologically diverse languages with 204K question-answer pairs. The data was collected from Wikipedia articles by human annotators. Unlike with SQuAD, the question writers formed questions without knowing the answers to them. The authors chose this strategy to reduce lexical overlapping between questions and passages, which could be exploited by machine learning systems.

One of the two datasets TyDi-QA consists of is in the SQuAD data format, which makes it ideal to combine with the SQuAD data. In total, it contains 7,635 Finnish questions. It is not much compared to SQuAD, but to the best of our knowledge, it is the only dataset that contains any Finnish data for extractive QA purposes. Consequently, we decided to include the Finnish partition of the TyDi-QA dataset in our experimental dataset.

3.4 The QA100-fi Corpus

Because most of the data used to train, validate, and test the models are synthetically generated, we decided to also create an additional small Finnish dataset for evaluation purposes only, QA100-fi. One option would have been to simply use the Finnish TyDi-QA data for evaluation. However, it would not have been feasible due to the possible differences with SQuAD questions caused by the TyDi-QA annotators not knowing the answers to their formed questions.

The QA100-fi dataset contains 100 questions related to Finnish Wikipedia articles. It is in the SQuAD format, and there are 10 questions for each category identified by Rajpurkar et al. (2016). We did not use any popularity-based ranking method to select the articles, like the authors of SQuAD did. Instead, we simply selected articles that appeared to be of good quality and had a length of at least three paragraphs. The dataset is tiny compared to actual QA test sets, but it still gives an impression of the models' performance on purely native text data collected by a native speaker.

3.5 Data Split

To train and evaluate models, we use data consisting of the answerable questions of the translated SQuAD1.1 data and the Finnish TyDi-QA data. Mimicking the methodology of Du et al. (2017), who used SQuAD data for English QG, we shuffled and split the data on article level into training, validation, and testing partitions. We call the resulting dataset SQuADTyDi-fi. The same SQuADTyDi-fi splits were used to train, validate, and evaluate both QA and QG models. We also use QA100-fi as an additional evaluation dataset. The split sizes are illustrated in Table 2.

Dataset	Split	Q-A Pairs	Articles
SQuADTyDi-fi	Train	64,604	6,977
	Dev	4,903	567
	Test	4,822	567
QA100-fi	Test	100	67

Table 2: Dataset splits. `Q-A Pairs` refers to the number of question-answer pairs in the corresponding split, and `Articles` tells how many Wikipedia articles the split has data from.

4 Model Fine-tuning

We train three models for QA and four models for QG. As the base models for fine-tuning, we use the Finnish GPT-2[2] (Radford et al., 2019), FinBERT[3] (Virtanen et al., 2019), and multilingual M-BERT, (Devlin et al., 2019).

[2] `https://huggingface.co/Finnish-NLP/gpt2-medium-finnish`

[3] We use `bert-base-finnish-cased-v1`, the cased variant.

4.1 BERT Question Answering

To use BERT for extractive QA, we employ the method described in Devlin et al., 2019. BERT is fine-tuned to "highlight" the answer when given a question and a passage that contains the answer as input. In practice, the model's task is to output two types of probabilities for each input token: **1)** being the answer span start **2)** being the last token of the answer span.

The input consists of a passage and a question, separated with the `[SEP]` token:

$$X = (\texttt{[CLS]},\ \langle \texttt{P} \rangle,\ \texttt{[SEP]},\ \langle \texttt{Q} \rangle) \quad (1)$$

where ⟨P⟩ is the input passage sequence and ⟨Q⟩ is the question sequence.

4.2 BERT Question Generation

The BERT models are fine-tuned for QG using the BERT-HLSQG (Highlight Sequential Question Generation) method originally presented by Chan and Fan, 2019b. In BERT-HLSQG, the previous decoding results are considered when decoding the next token. Tokens are generated one by one using a strategy to modify BERT into generating text in an autoregressive manner. Another key idea in HLSQG is to highlight the answer in the input passage with special tokens to tackle any ambiguity caused by the answer appearing multiple times in the passage.

At inference, the input X for an HLSQG model is in the following format:

$$X = (\texttt{[CLS]}, P_{HL}, \texttt{[SEP]}, \hat{Q}, \texttt{[MASK]}) \quad (2)$$

where P_{HL} is the highlighted passage sequence and $\hat{Q}$ is the predicted question sequence.

At the first inference step, the highlighted passage is followed only by a `[MASK]` token, as the predicted question sequence $\hat{Q} = [\hat{q}_1, \hat{q}_2, ..., \hat{q}_{|\hat{Q}|}]$ is empty at the start. The passage highlighting is done by placing special `[HL]` tokens around the answer in the passage:

$$P_{HL} = (p_1, ..., \texttt{[HL]}, p_s, ..., p_e, \texttt{[HL]}, ..., p_{|P|}) \quad (3)$$

where p_n is the *n*th passage token, p_s and p_e are the answer start and end tokens, and $|P|$ is the passage length.

During each step, the whole input is fed to the model, and it outputs a prediction for the `[MASK]` token. That prediction is considered the next token in the question sequence, and a new `[MASK]` token is placed after it. The same procedure goes on with inputs updated with the newly predicted question tokens until a `[SEP]` token is predicted. At that point, the question is considered ready.

4.3 GPT-2 Question Answering

To fine-tune a GPT-2 model for QA (`GPT-2-QA`), we use a prompt to encourage the model to generate answers relevant to the given passage and question. The model should learn the pattern of the prompt and also the relation between the two input sections (passage and question) in the prompt.

During fine-tuning, the prompt consists of three lines. Each line starts with a word that describes the content of the line and is followed by a matching sequence. For example, the first two lines start with *Context:* and *Question:* and continue with the passage and question sequences. During training, language modeling loss is computed only on the section where the model should output the answer. The fine-tuning prompt is:

$$X = \\ Context{:}\langle \texttt{P} \rangle \quad Question{:}\langle \texttt{Q} \rangle \quad Answer{:}\langle \texttt{A} \rangle$$

where ⟨P⟩ is the passage sequence, ⟨Q⟩ is the question sequence, and ⟨A⟩ is the answer sequence. During inference, the answer sequence is omitted from the prompt, as the model's task is to fill it in.

4.4 GPT-2 Question Generation

We train two GPT-2-based QG models, `GPT-2-QG` and `GPT-2-HLQG`. The training and inference prompts of the `GPT-2-QG` model are the same as the `GPT-2-QA`, but the order of the last two rows is reversed. The QG models should learn to use the passage to generate a question that the second line's sequence answers. The training procedure is the same as with `GPT-2-QA`, but instead of answers, the training loss is computed on the generated questions. The two QG models differ in the prompts. `GPT-2-HLQG` also highlights the answer in the passage with `[HL]` tokens. The motivation for that is the same as with BERT-HLSQG: to reduce the possible ambiguity caused by the answer appearing multiple times in the passage.

4.5 Implementation

All the pre-trained models were accessed via the `transformers`[4] Python package by Hugging

[4] `https://github.com/huggingface/transformers`. Version `3.0.2` for BERT-HLSQG

Face (Wolf et al., 2020). The fine-tuning scripts were implemented using the same package along with PyTorch.[5]. For fine-tuning BERT-HLSQG models, we modified and used open-source code by Lin (2020).[6]

We fine-tune the models using two Nvidia Volta v100 GPUs and AdamW optimization with initial learning rate 5×10^{-5}. The batch size varied from 2 to 24, depending on the task and the model architecture. All the models were trained for six epochs, and a validation set was used to keep track of the training performance and thus select the best model for evaluation on the test sets. QA BERT models (`FinBERT-QA` and `M-BERT-QA`) had the best validation results after two epochs, whereas all the other models had the best validation performance after six epochs. More details regarding the fine-tuning are included in Appendix A.

5 Results

5.1 QA Results

The evaluation results for the QA models are in Table 3. The scores are multiplied by 100 to mimic the style of the official SQuAD leaderboard.[7] With both testing datasets, `FinBERT-QA` obtains the best results. However, the fine-tuned M-BERT model comes close, with EM scores 2-3% worse and F1 scores 2.8-4.5 points behind FinBERT-QA. The GPT-2 -based QA model achieves moderately good results also, but both EM and F1 scores are at least 20 points worse with both test sets.

Dataset	Model	Exact Match	F1 score
SQuADTyDi-fi	`FinBERT-QA`	**58.0**	**69.9**
	`M-BERT-QA`	56.0	67.1
	`GPT-2-QA`	37.2	46.9
QA100-fi	`FinBERT-QA`	**67.0**	**83.7**
	`M-BERT-QA`	64.0	79.2
	`GPT-2-QA`	43.0	56.0

Table 3: Evaluation of QA models on two test sets.

`GPT-2-QA` model obtained the worst results on both datasets. With an EM score of 37.2 and an F1 score of 46.9 on SQuADTyDi-fi data, it is apparent that fine-tuning has contributed to the model's ability to answer questions. The model outputs relatively short answers as expected, and it also seems to have quite well learned the expected answer type for each interrogative in the question. For example, the model mostly seems to answer questions starting with *kuka* ("who") with names/people and questions starting with *montako* ("how many") with numeral phrases. However, the results are still far behind the best-performing models.

When the question contains very different vocabulary than the passage (e.g., synonyms or idiomatic expressions), `GPT-2-QA` seems to perform particularly poorly. A closer look at the results shows that the `GPT-2-QA` model's outputs occasionally contain words that are slightly modified versions of the ones in the passage. This problem is unique to GPT-2 in the experiments as it is the only autoregressive model. Some other examples of such errors are shown in Table 4. However, most of the answers seem to be substrings of the input passages, as expected. `GPT-2-QA` seems to often fail to "understand" what specifically is being asked. Even when it seems to understand that the question should be answered with a date and the answer should be a substring of the passage, it often seems to pick just any date. And sometimes, it even modifies the date, as seen in Table 4.

Predicted answer	Target answer
Kenji Vatanabe	Kenji Watanabe
20. lokakuuta 2000	21. lokakuuta 2000
Kypylän	Midnan kypärän
3 vuotta	kolme vuotta

Table 4: Examples of `GPT-2-QA` outputs that are not substrings of the input passage.

The other QA models, `FinBERT-QA` and `M-BERT-QA`, perform much better. They come in quite close to each other as `FinBERT-QA` outperforms `M-BERT-QA` by 2-3 points on SQuADTyDi-fi data with its EM and F1 scores of 58.0 and 69.9, respectively. The difference between the scores of `FinBERT-QA` and `M-BERT-QA` is slightly bigger with the QA100-fi test data, with which `FinBERT-QA` obtains an EM score of 67.0 and an F1 score of 83.7. Using only Finnish data and a lot larger amount of it in pre-training seems to have been beneficial for `FinBERT-QA`. Like

models and `4.8.1` for other models.

[5]Version `1.5.0+cu101` for BERT-HLSQG models and `1.9.0+cu111` for other models.

[6]`https://github.com/chris4540/StudyMaskedLMForQG`

[7]`https://rajpurkar.github.io/SQuAD-explorer/`

GPT-2-QA, also M-BERT-QA seems to occasionally struggle when the question is phrased very differently compared to the input passage.

As with GPT-2-QA, the longer the ground truth answer, the more likely the BERT-based models seem to predict it incorrectly. However, rather than choosing a completely wrong span, FinBERT-QA and M-BERT-QA often seemed only to pick too few words. This is also reflected in the bigger differences between EM and F1 scores of the other two models, compared to GPT-2-QA. Other than questions with longer answers, it is challenging to identify any specific question/answer types with which FinBERT-QA and M-BERT-QA have the most difficulties. Additional examples of outputs of the QA models are included in Appendix A.

The results of all QA models are better with the QA100-fi test dataset. It is possible that because the passages, questions, and answers in QA100-fi are not machine-translated, they could be closer to the Finnish language with which the models were pretrained. Another factor might be the lengths of the passages, questions, and answers. Their average lengths are shown in Table 5. The passages and questions in the test partition of SQuADTyDi-fi are longer on average, but the answers are longer in QA100-fi. Longer passages are more challenging for the models as there are more tokens from which to choose the answer span start and end tokens. However, the test sets are so different in size that it is hard to say how much that affects the results.

	Passage	Question	Answer
SQuADTyDi-fi (test)	74.5	6.6	2.5
QA100-fi	62.2	5.9	3.2

Table 5: Average word counts in the test partition of SQuADTyDi-fi and QA100-fi.

As there are no other Finnish QA models to compare with, we can gain some perspective by comparing the results with English models trained on a similar dataset. The top EM and F1 scores for single BERT models in the English SQuAD1.1 leaderboard[8] are around 85 and 90, respectively. The overall best single model results are from other transformer-based models, like LUKE (Yamada et al., 2020) and XLNet (Yang et al., 2019), which both obtain EM and F1 scores over 90 and 95, respectively. The best Finnish results (by FinBERT-QA) are quite far from the best-performing English models. However, it is worth noting that the Finnish models were fine-tuned using a smaller dataset which is probably of poorer quality, as it has been automatically translated. Finnish being a highly inflective language might also make the QA task generally more challenging.

[8]Webpage mirroring SQuAD1.1 leaderboard: https://paperswithcode.com/sota/question-answering-on-squad11

5.2 QG Results

The evaluation results for the QG models are in Table 6. The FinBERT-based models obtain the best results. As in the QA task, the results of the FinBERT and M-BERT-based models are quite close to each other, whereas the GPT-2 models are much worse.

Dataset	Model	BLEU-4	METEOR
SQuADTyDi-fi	FinBERT-HLSQG	**0.11**	**0.17**
	M-BERT-HLSQG	0.10	0.16
	GPT-2-QG	0.04	0.10
	GPT-2-HLQG	0.04	0.10
QA100-fi	FinBERT-HLSQG	**0.18**	**0.22**
	M-BERT-HLSQG	0.13	0.20
	GPT-2-QG	0.04	0.13
	GPT-2-HLQG	0.04	0.11

Table 6: BLEU-4 and METEOR scores of QG models. Results on additional metrics in Appendix A.

Both GPT-2-QG and GPT-2-HLQG achieve a BLEU-4 score of 0.04 on both datasets. Unlike in Chan and Fan (2019b), using an answer highlight technique in the passage did not lead to an increase in the performance as the results of the two models are nearly identical. This indicates that ambiguity was not the root cause of the inferior performance of the models.

Looking at the outputs of the GPT-2-based QG models, it is clear that the models learn the general structure of a question. The outputs mostly start with the correct interrogative word and end with a question mark. The questions also seem mostly grammatical. The biggest problems seem to be related to semantic validity and generating questions that can be answered using the input answer. However, the models occasionally seem to generate questions that can be answered with the input answer, but they are very different from the ground-truth questions. They are good examples

of why using automatic, n-gram-based evaluation metrics to assess QG systems can be problematic.

Compared to the GPT-2-based QG models, the BERT-based QG models perform roughly twice as well on every metric. `FinBERT-HLSQG` and `M-BERT-HLSQG` seem to output questions that make more sense and have more common words with the target question. For example, with target question *Kuinka korkeaksi puu yleensä kasvaa avoimilla alueilla?* ("How tall does the tree usually grow in open areas?"), `FinBERT-HLSQG` outputs *Minkä korkuinen on jousisoihtupuu avoimilla alueilla?* ("How tall is the pink trumpet tree in open areas?") and `GPT-2-HLQG` outputs *Minkä kokoisia puutalot ovat metsäalueiden korkeilta tasoilta?* ("What size are the wooden houses from the high levels of the forest areas?"). `GPT-2-HLQG`'s output is nonsensical yet grammatical, whereas `FinBERT-HLSQG`'s output can be considered correct, though the phrasing is quite different from the target question. All models perform better with shorter passages and struggle at inflecting rare words. Additional examples of the outputs of all QG models are shown in Appendix A.

As on the QA task, the FinBERT-based model achieves slightly better scores on the SQuADTyDi-fi test set than the multilingual variant. However, in QG, the difference between the performance of BERT-based models is bigger when evaluating on the QA100-fi dataset. For example, `FinBERT-HLSQG` obtains a BLEU-4 score of 0.18, while `M-BERT-HLSQG` yields 0.13. Checking the outputs on QA100-fi, it seems that `M-BERT-HLSQG` has more problems inflecting words, and it occasionally uses word order and phrasings that sound a bit unnatural in Finnish. It is possible that these problems were exacerbated when the model was tested on QA100-fi, which consists of data collected by a native speaker.

Chan and Fan (2019b), who initially presented the BERT-HLSQG method, report a BLEU-4 score of 0.20 for their English QG model that was fine-tuned on roughly 73K question-answer pairs. `FinBERT-HLSQG`'s BLEU-4 score (0.11) on the SQuADTyDi-fi test set is quite far from that, whereas the BLEU-4 score on the smaller QA100-fi test set (0.18) is a lot closer. It is likely that the passages and questions in QA100-fi being shorter on average has a positive effect on the model's performance on the dataset. Chan and Fan (2019a) also conclude that their BERT-HLSQG model works better with shorter passages. As with the QA task, it is possible that the smaller amount of training data and its poorer quality, together with the more complex Finnish morphology, partly explain the differences that occur when compared to the English models.

6 Conclusion and Future Work

We have proposed an MT-based method for creating a Finnish QA dataset, and used it to train and evaluate several transformer-based QA and QG models. On both tasks, fine-tuned monolingual BERT models obtain the best results. The multilingual variants came close, while the fine-tuned GPT-2 models were found to underperform. Pretraining with only Finnish data seems to give the models an edge in both QA and QG.

To the best of our knowledge, these are the first monolingual Finnish QA and QG models. They set a fair baseline for further research in Finnish QA and QG. All data used in the experiments is released to the research community, to support future research, and the models are released as benchmarks. We believe that this is a valuable contribution, since suitable datasets created by native Finnish speakers are not yet available.

Given the promising initial results, we plan to pursue several directions. (1) As the SQuAD2.0 data with the unanswerable questions was also translated, it could be used to train the first Finnish QA models that can also identify unanswerable questions. (2) Lower-level natural language processing (NLP) components can be employed to study and improve performance. For example, we can use syntactic parsing to check for ungrammatical questions, to analyze the created synthetic dataset; we can use name recognition to improve QA results (Yadav and Bethard, 2019; Piskorski et al., 2019), etc. (3) Real-world applications, such as language learning systems, e.g., (Katinskaia et al., 2018, 2017), can benefit from QA and QG—by automatically generating reading comprehension questions from arbitrary authentic text. To integrate QG into such applications, a separate model should be developed for choosing the appropriate input answers. (4) To support (3), it is important to study in detail on what types questions and answers the QA and QG models do especially well or especially poorly.

References

Negin Abadani, Jamshid Mozafari, Afsaneh Fatemi, Mohammd Ali Nematbakhsh, and Arefeh Kazemi. 2021. Parsquad: Machine translated SQuAD dataset for Persian question answering. In *2021 7th International Conference on Web Research (ICWR)*, pages 163–168.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Lili Aunimo, Juha Makkonen, and Reeta Kuuskoski. 2004. Cross-language question answering for Finnish. In *Proceedings of the Web Intelligence Symposium, Finnish Artificial Intelligence Conference*, pages 35–49.

Casimiro Pio Carrino, Marta R. Costa-jussà, and José A. R. Fonollosa. 2020. Automatic Spanish translation of SQuAD dataset for multi-lingual question answering. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5515–5523, Marseille, France. European Language Resources Association.

Ying-Hong Chan and Yao-Chung Fan. 2019a. BERT for question generation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 173–177, Tokyo, Japan. Association for Computational Linguistics.

Ying-Hong Chan and Yao-Chung Fan. 2019b. A recurrent BERT-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162, Hong Kong, China. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470. ArXiv: 2003.05002.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin d'Hoffschmidt, Wacim Belblidia, Quentin Heinrich, Tom Brendlé, and Maxime Vidal. 2020. FQuAD: French question answering dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1193–1208, Online. Association for Computational Linguistics.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Ali Kabbadj. 2018. Something new in French text mining and information extraction (universal chatbot): Largest Q&A French training dataset (110 000+). [Online; posted 11-November-2018].

Anisia Katinskaia, Javad Nouri, and Roman Yangarber. 2017. Revita: a system for language learning and supporting endangered languages. In *Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition*, pages 27–35, Gothenburg, Sweden. LiU Electronic Press.

Anisia Katinskaia, Javad Nouri, and Roman Yangarber. 2018. Revita: a language-learning platform at the intersection of ITS and CALL. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

T. Klein and Moin Nabi. 2019. Learning to answer by learning to ask: Getting the best of GPT-2 and BERT worlds. *ArXiv*, abs/1911.02365.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics. ArXiv: 1910.13461.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. KorQuAD1.0: Korean QA dataset for machine reading comprehension. *arXiv:1909.07005 [cs]*. ArXiv: 1909.07005.

Chun Hung Lin. 2020. *Automatic Question Generation with Pre-trained Masked Language Models*. Ph.D. thesis, KTH Royal Institute of Technology, Stockholm, Sweden.

Susumu Okazawa. 2021. Swedish translation of SQuAD2.0. GitHub repository (Accessed: 6 March 2022).

Jakub Piskorski, Laska Laskova, Michał Marcińczuk, Lidia Pivovarova, Pavel Přibáň, Josef Steinberger, and Roman Yangarber. 2019. The second cross-lingual challenge on recognition, normalization, classification, and linking of named entities across Slavic languages. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*. ACL.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8).

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for Finnish. *arXiv:1912.07076 [cs]*. ArXiv: 1912.07076.

Bingning Wang, Xiaochuan Wang, Ting Tao, Qi Zhang, and Jingfang Xu. 2020. Neural question generation with answer pivot. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9138–9145. Number: 05.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. Technical Report arXiv:1910.03771, arXiv. ArXiv:1910.03771 [cs] type: article.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144 [cs]*. ArXiv: 1609.08144.

Vikas Yadav and Steven Bethard. 2019. A survey on recent advances in named entity recognition from deep learning models. *CoRR*, abs/1910.11470.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Zhuosheng Zhang, Junjie Yang, and Hai Zhao. 2020. Retrospective Reader for Machine Reading Comprehension. Technical Report arXiv:2001.09694, arXiv. ArXiv:2001.09694 [cs] type: article.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. *arXiv:1704.01792 [cs]*. ArXiv: 1704.01792.

A Appendix

Model	Epochs (best model)	Batch size
FinBERT-QA	2	16
M-BERT-QA	2	16
GPT-2-QA	6	2
FinBERT-HLSQG	6	24
M-BERT-HLSQG	6	16
GPT-2-QG	6	2
GPT-2-HLQG	6	2

Table 7: Training hyperparameters. With all models, we use the AdamW optimization algorithm with an initial learning rate of 5×10^{-5}.

Dataset	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
SQuADTyDi-fi	FinBERT-HLSQG	**0.29**	**0.21**	**0.15**	**0.11**	**0.17**	**0.33**
	M-BERT-HLSQG	0.29	0.20	0.14	0.10	0.16	0.31
	GPT-2-QG	0.18	0.11	0.06	0.04	0.10	0.20
	GPT-2-HLQG	0.18	0.10	0.06	0.04	0.10	0.20
QA100-fi	FinBERT-HLSQG	**0.39**	**0.30**	**0.22**	**0.18**	**0.22**	**0.41**
	M-BERT-HLSQG	0.36	0.25	0.18	0.13	0.20	0.36
	GPT-2-QG	0.22	0.12	0.07	0.04	0.13	0.22
	GPT-2-HLQG	0.19	0.11	0.07	0.04	0.11	0.20

Table 8: All evaluation results of the QG models.

Input passage	Ulkomuodoltaan hylkeet ovat sileitä ja pulleita. Ruumiinrakenne soveltuu sulavaan vedessä liikkumiseen. Ranteesta ja kämmenestä ovat muodostuneet etuevät ja nilkasta ja jalkaterästä takaevät. Evät ovat heikot eikä niitä voi käyttää apuna maalla liikkumiseen . Hylkeet liikkuvatkin maalla siten, että ne siirtävät painoa rinnan ja vatsan varaan. Erotuksena lähisukulaisistaan korvahylkeistä, joihin kuuluvat muun muassa merileijonat, varsinaisilla hylkeillä ei ole ulkoisia korvalehtiä. Varsinaisten hylkeiden uiminen tapahtuu evien ja ruumiin takaosan sivuttaissuuntaista liikettä apuna käyttäen.
Input question	Mihin hylkeiden evät eivät sovellu? (*What are seal fins not suitable for?*)
Target answer	maalla liikkumiseen (*to move on land*)
Model	**Predicted Answer**
FinBERT-QA	maalla liikkumiseen. (*to move on land.*)
M-BERT-QA	vedessä (*in the water*)
GPT-2-QA	ui maalla (*swim/swims on land*)

Table 9: Output examples of the QA models. The ground truth answer is highlighted in the input passage.

Input passage	Jättiläismetsäkarju eli jättiläismetsäsika eli jättisika (Hylochoerus meinertzhageni) on keskisen ja läntisen Afrikan metsissä elävä elinvoimainen sorkkaeläinlaji. Se on sukunsa Hylochoerus ainoa laji. Jättiläismetsäkarjut ovat suurimpia luonnonvaraisia sikoja. Ne voivat kasvaa jopa 210 senttimetriä pitkiksi ja painaa 275 kilogrammaa. Niiden ruumis on tanakka ja pää leveä, mutta jalat ovat lyhyet. Nahkaa peittävät pitkät ja karkeat karvat, jotka nousevat pystyyn eläimen kiihtyessä.
Input answer	210 senttimetriä *(210 centimeters)*
Target question	Kuinka pitkiksi jättiläismetsäkarjut voivat kasvaa? *(How long can giant forest hogs grow?)*

Model	**Generated question**
`FinBERT-HLSQG`	Kuinka pitkäksi jättiläismetsäkarju voi kasvaa? *(How long can a giant forest hog grow?)*
`M-BERT-HLSQG`	Kuinka pitkiä jättiläismetsäkarjat voivat kasvaa? * *(How long can giant forest cattles grow?)*
`GPT-2-QG`	Miten pitkäksi afrikkalainen jättiläismetsäkarju voi kasvaa? *(How long can an African giant forest hog grow?)*
`GPT-2-HLQG`	Kuinka pitkä on jättiläismetsäkarjun pituus? *(How long is the length of a giant forest hog?)*

Table 10: Output examples from the QG models. The input answer is highlighted in the input passage. Outputs marked with * contain inflection errors, but they are ignored in the translation.

Probing structural constraints of negation in Pretrained Language Models

David Kletz[1,2] and **Marie Candito**[1] and **Pascal Amsili**[2]
(1) Université Paris Cité & LLF (CNRS/UPC)
(2) Université Sorbonne Nouvelle & Lattice (CNRS/ENS-PSL/USN)
david.kletz@sorbonne-nouvelle.fr, marie.candito@u-paris.fr, pascal.amsili@ens.fr

Abstract

Contradictory results about the encoding of the semantic impact of negation in pretrained language models (PLMs) have been drawn recently (e.g. Kassner and Schütze (2020); Gubelmann and Handschuh (2022)). In this paper we focus rather on the way PLMs encode negation and its formal impact, through the phenomenon of the Negative Polarity Item (NPI) licensing in English. More precisely, we use probes to identify which contextual representations best encode 1) the presence of negation in a sentence, and 2) the polarity of a neighboring masked polarity item. We find that contextual representations of tokens inside the negation scope do allow for (i) a better prediction of the presence of *not* compared to those outside the scope and (ii) a better prediction of the right polarity of a masked polarity item licensed by *not*, although the magnitude of the difference varies from PLM to PLM. Importantly, in both cases the trend holds even when controlling for distance to *not*. This tends to indicate that the embeddings of these models do reflect the notion of negation scope, and do encode the impact of negation on NPI licensing. Yet, further control experiments reveal that the presence of other lexical items is also better captured when using the contextual representation of a token within the same syntactic clause than outside from it, suggesting that PLMs simply capture the more general notion of syntactic clause.

1 Introduction

Negation has recently been the focus of various works aiming at determining the abilities of Pretrained Language Models (PLMs) to capture linguistic knowledge.

Some works investigate the 'semantic impact' of negation, namely its impact in terms of truth values, by interpreting how the presence of negation impacts the probability distribution at a masked position. The rationale is that negating a verb reverses the truth value of its clause, which should be reflected in the probability distribution at certain positions. Ettinger (2020); Kassner and Schütze (2020) use factual statements such as (1), and report that models output similar distributions for the positive and negative variants of (1), and conclude that models largely ignore negation.

(1) *A robin is (not) a [MASK]*

Gubelmann and Handschuh (2022) chose to avoid factual statements and to focus rather on multi-sentence self-contained examples, such that, given the context provided by the first sentence, one particular word is either likely (in positive items) or ruled out (in negative items) at a masked position in the second sentence. Because this particular word is substantially less often the top-1 prediction in the negative items than in the positive items, the authors draw the opposite conclusion that PLMs do show sensitivity to negation.

A different line of works focused on finding out to what extent negation is encoded in PLM embeddings. Celikkanat et al. (2020) train classifiers taking as input the contextual embedding of a verb or its subject or direct object, and predicting whether the verb is negated or not. The resulting high accuracy allows them to conclude that these tokens' embeddings do contain "traces" of *not*. More generally, several authors have investigated whether the contextual representation of a token encodes information about surrounding tokens. To ease further reading, we will talk of a classifier taking as input an **input embedding**, i.e. the contextual

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 541–554

representation of an **input token**, and predicting some **target information** about another token in the sentence. For instance, Klafka and Ettinger (2020) study how input embeddings encode animacy, gender, and number of surrounding words in a specific SVO context. Li et al. (2022) target the number feature of French participles in the context of object-past participle agreement. They show that the performance of the classifier depends on the syntactic position of the input token in the sentence. We will build on their idea to compare performance at predicting target information depending on the syntactic zone the input token belongs to. In this paper, one of the probed target information will be the presence or absence of a given word within the sentence, which we call the **target token**.

More precisely, our aim is to study PLMs' ability to capture and encode structural information concerning negation (namely negation scope). To do so we first probe whether input embeddings can serve to accurately predict the presence or absence of a target *not*.[1] Moreover, we wish to test PLMs' ability to actually *mobilize* this encoding to capture phenomena that are direct consequences of the presence of negation. To do so, we focus on the licensing of Negative Polarity Items (NPI) by *not* modifying a verb. Polarity Items (PI), either positive (e.g. *some*), or negative (e.g. *any*), are words or expressions that are constrained in their distribution (Homer, 2020). A NPI will require that a word or a construction, called the **licensor**, be in the vicinity. More precisely, the licensor itself grammatically defines a zone of the sentence, called the **licensing scope**, in which the NPI can appear. The adverb *not* modifying a verb is one such licensor. While *any* is licensed by negation in (2-a) *vs.* (2-b), it is not licensed in (2-c), even though the verb is negated, arguably because it is not in the licensing scope[2].

(2) a. Sam didn't find any books.
b. *Sam found any books.
c. *Any book was not found by Sam.

Jumelet and Hupkes (2018) have shown that LSTM embeddings do encode the notion of licensing scope (given an input embedding, a classifier can predict the structural zone the input token belongs to), a finding later confirmed for transformer-based PLMs (Warstadt et al., 2019). Focusing on when the licensor is a verb-modifying *not*, we rather investigate whether this encoding of the zones *go as far as* enabling a better prediction of a PI's polarity from *inside* the licensing scope compared to *outside* the scope. So instead of the question *"Is this input embedding the embedding of a token located within, before or after the licensing scope?"*, we rather ask the question *"Given a masked PI position, and an input embedding of a neighboring token, what is the polarity of the PI?"*, and we study whether this question is better answered when the input embedding is inside or outside the licensing or negation scopes.

Note that our methodology differs from that of Jumelet and Hupkes (2018), who, given an input token, predict the zone this token belongs to. We instead predict the polarity of a neighboring masked polarity item and then compare accuracies depending on the input token's zone. Our motivation is that the polarity, being a lexical information, requires less linguistic preconception, and hence our probing method is a more direct translation of the NPI licensing phenomenon: we study whether and where the information of "*which PIs are licit where?*" is encoded, in the context of sentence negation. This method also allows us to better control the confounding factor of distance between the input embedding and the licensor *not*.

In the following, we define the linguistic notions of negation scope and NPI licensing scope in section 2, and show how we actually identified them in English sentences. In section 3, we describe our probing experiments and discuss their results, both for the encoding of *not* (section 3.1), and the encoding of NPI licensing (section 3.2). We then study the more general ability of PLMs to deal with clause boundaries (section 4), and conclude in section 5.

2 Defining and identifying scopes

2.1 Negation scope

From a linguistic point of view, the scope of a negation cue is the area of the sentence whose propositional content's truth value is reversed by the presence of the cue. While in many cases it is sufficient to use the syntactic structure to recover the scope, in some cases semantics or even prag-

[1] We restrict our probing to *not*, which is by far the most frequent negation clue (57% of the occurrences, while the second most frequent, *no*, accounts for 21% of occurrences).

[2] We leave aside the uses of *any* and the like having *free choice* interpretations, as for instance in *"Pick any card"*.

matics come into play.[3] Nevertheless, annotation guidelines usually offer syntactic approximations of negation scope.

To identify the negation scope for *not*[4] modifying a verb, we followed the syntactic constraints that can be inferred from the guidelines of Morante and Blanco (2012). Note though that these guidelines restrict the annotation to factual eventualities, leaving aside e.g. negated future verbs. We did not retain such a restriction, hence our identification of the negation scope is independent from verb tense or modality.

2.2 NPI licensing scope

Polarity items are a notoriously complex phenomenon. To identify the NPI licensing scope, we focus on specific syntactic patterns defined by Jumelet and Hupkes (2018), retaining only those involving *not* as licensor.[5] Table 1 shows an example for each retained pattern (hereafter the **neg-patterns**), with the NPI licensing scope in blue.

Importantly, in the neg-patterns, the licensing scope is strictly included in the negation scope: within the clause of the negated verb, the tokens to its left belong to the negation scope but not to the licensing scope. E.g. in (3), *anyone* is not licit as a subject of *going*, whether the location argument is itself a plain PP, a NPI or a PPI (3-b).

(3) a. I'm not going anywhere.
b. *Anyone is not going to the party/somewhere/anywhere.

We thus defined 4 zones for the *not*+NPI sentences, exemplified in Table 1: **PRE** (tokens before both scopes), **PRE-IN** (to the left of the licensing scope, but within the negation scope), **IN** (in both scopes), and **POST** (after both scopes).

We note though that the restriction exemplified in (3-b) only holds for non-embedded NPIs (de Swart, 1998), so examples like (4), with an embedded NPI in the subject of the negated verb (hence belonging to our **PRE-IN** zone), are theoretically possible.

(4) Examples with any relevance to that issue didn't come up in the discussion.

Yet in practice, we found that they are extremely rare: using the Corpus of Contemporary American English (COCA, Davies 2015)[6], we extracted sentences matching one of the neg-patterns, and among these, sentences having *any* or *any-body/one/thing/time/where* in the IN zone, the PRE-IN zone or both. As shown in Table 2, *any** in the PRE-IN zone are way rarer than in the classical licensing scope (IN zone)[7]. Hence we sticked to the usual notion of direct NPI licensing scope, as illustrated in Table 1.

2.3 Building the not+NPI test set

Having defined these structural zones, we could use them to probe the traces they carry and compare the magnitude of these traces across the four zones. To do so, we built a test set of COCA sentences containing *not* licensing a NPI (hereafter the ***not*+NPI** test set), matching one of the neg-patterns of Table 1, and having at least one *any, anybody, anyone, anything, anytime* or *anywhere* within the licensing scope.

The scope of negation has been implemented through an approximation using dependency parses (from the Stanza parser (Qi et al., 2020)), which proved more convenient than phrase-structure parses: we took the subtree of the negated verb, excluding *not* itself, and excluding dependents corresponding to sentential or verbal conjuncts and to sentential parentheticals.

More precisely, we identified the token having *not* as dependent (which, given our patterns, can be either the negated verb or a predicative adjective in case of a negated copula). Then, we retrieved the children of this head, except those attached to it with a "conj", "parataxis", "mark" or "discourse" dependency. In the complete subtrees of the selected dependents, all tokens were annotated as being inside the negation scope.

[3] For instance in *Kim did not go to the party because Bob was there.*, negation may scope only over the matrix clause or include the causal subordinate clause.

[4] In all this article, *not* stands for either *not* or *n't*.

[5] We ignored pattern 4 (*never* instead of *not* as licensor), and 6 (too few occurrences in our data). We merged patterns 1 and 2, and corrected an obvious minor error in pattern 5.

[6] We used a version with texts from 1990 to 2012. COCA is distributed with some tokens in some sentences voluntarily masked, varying across distributions. We ignored such sentences.

[7] More precisely, the figures in Table 2 correspond to an upper bound, because of (i) potential syntactic parsing errors impacting the identification of the zones, (ii) cases in which the NPI licensor is different from the *not* targeted by the patterns, and (iii) cases in which *any** is a free choice item rather than a NPI.We inspected 250 examples of *any** in the PRE-IN zone, and 250 examples in the IN zone. In the former, we found that almost all cases fall under (i), (ii) or (iii), less than 3% corresponding to examples such as (4)). In contrast, in the IN zone the proportion of NPIs actually licensed by the target *not* is 92%.

Id	Pattern	Example and zones
1/2	(VP (VB*/MD) (RB not) VP)	I have my taxi and I 'm not going anywhere but my brother will leave Spain because he has a degree.
3	(VP (VB*) (RB not) NP/PP/ADJP)	Since it is kind of this fairy-tale land, there aren't any rules of logic so you can do anything, she says.
5*	(S (RB not) VP)	I went in early, not wanting anyone to see me and hoping for no line at the counter.

Table 1: The "**neg-patterns**": patterns adapted from Jumelet and Hupkes (2018), which we used to identify some cases of *not* licensing a NPI and to build the *not*+NPI test set. **Col1**: pattern id in Jumelet and Hupkes (2018). **Col2**: syntactic pattern (defined as a phrase-structure subtree, using the Penn Treebank's annotation scheme), with the licensing scope appearing in blue. **Col3**: examples with colors for the four zones: pink for tokens in the PRE zone (before both scopes), purple for PRE-IN (to the left of the licensing scope, but within the negation scope), blue for IN (within both scopes) and green for POST (after both scopes). The NPI licensor is *not*, and appears in yellow.

Total	IN	PRE-IN	both
45,157	35,938	711	58

Table 2: Number of sentences from the COCA corpus, matching the neg-patterns of Table 1: **Col1**: total number, **Col2-4**: number having *any** in the IN zone, the PRE-IN zone, and in both zones respectively.

with *not*	2,285,000
↪ with *NPI*	143,000
↪ pattern 1	30,896
↪ pattern 3	2,529
↪ pattern 5	1,020
↪ pattern 6	< 100

Table 3: Statistics of the *not*+NPI test set: number of COCA sentences matching the neg-patterns (cf. Table 1), and having at least one *any** in the IN zone (licensing scope).

For the licensing scope, we parsed the corpus using the PTB-style parser "Supar Parser"[8] of Zhang et al. (2020), and further retained only the sentences (i) matching at least one of the neg-pattern of Table 1 and (ii) having a NPI within the licensing scope (IN zone, shown in blue in Table 1), resulting in the *not*+NPI test set, whose statistics are provided in Table 3.

3 Probing for the scopes

Our objective is to study how a transformer-based PLM (i) encodes the presence of a negation (the "traces" of negation) and (ii) models lexico-syntactic constraints imposed by negation, such as the modeling of a NPI licensing scope. Using the terminology introduced in section 1, we probe whether input embeddings encode as target information (i) the presence of *not* elsewhere in the sentence, and (ii) the polarity of a masked PI. The former focuses on a plain encoding of negation, whereas the latter focuses on whether the encoding of negation can be mobilized to reflect a property (NPI licensing) that is directly imposed by negation. To investigate whether such an encoding matches linguistic notions of scopes, we contrast results depending on the zone the input token belongs to (among the four zones defined for *not* licensing a NPI, namely PRE, PRE-IN, IN, POST) and its distance to *not*.

We studied four PLMs : BERT-base-case, BERT-large-case (Devlin et al., 2019) and ROBERTA-base and ROBERTA-large (Liu et al., 2019). All our experiments were done with each of these models, and for a given model, each experiment was repeated three times. All the sentences we used for training, tuning and testing were extracted from the COCA corpus.

3.1 Probing for the negation scope

In preliminary experiments, we extended Celikkanat et al. (2020)'s study by investigating the traces of *not* in the contextual embedding of all the tokens of a sentence containing *not* (instead of just the verb, subject and object).

3.1.1 Training neg-classifiers

We trained binary classifiers (hereafter the **m-neg-classifiers**, with *m* the name of the studied PLM) taking an input contextual embedding, and predicting the presence or absence of at least one *not* in the sentence. In all our experiments, the PLMs parameters were frozen. We trained 3 classifiers for each of the 4 tested PLMs. To train and

[8] https://parser.yzhang.site/en/latest/index.html

evaluate these classifiers, we randomly extracted 40,000 sentences containing exactly one *not*, and 40,000 sentences not containing any *not*. These sentences were BERT- and ROBERTA-tokenized, and for each model, we randomly selected one token in each of these sentences to serve as input token. Among these input tokens, we ignored any token *not*, as well as all PLM tokens associated to a contracted negation: for instance *don't* is BERT-tokenized into *don* + ' + *t*, and ROBERTA-tokenized into *don'* + *t*. These tokens were ignored since they are too obvious a clue for the presence of a verbal negation. Furthermore, in order to homogenize the handling of negation whether contracted or not, we also set aside any modal or auxiliary that can form a negated contracted form. Hence, in *She did leave*, *She did not leave* or *She didn't leave*, the only candidate input tokens are those for *She* and *leave*[9]. We used 64k sentences for training (**neg-train-sets**), and the remaining 16k for testing (**neg-test-set**).

We provide the obtained accuracies on this neg-test-set in Table 4, which shows that performance is largely above chance. We provide a more detailed analysis of the classifers performance in section 3.2.

Model	BERT$_b$	BERT$_l$	ROB.$_b$	ROB.$_l$
Accur.	74.3	73.1	72.1	76.6

Table 4: Accuracies of the neg-classifiers on the neg-test-set for each PLM (averaged over 3 runs).

3.1.2 Studying results on the *not*+NPI test set

To probe the negation scope, we then used the *not*+NPI test set (cf. section 2), and compare accuracies in PRE-IN vs. PRE, and in IN vs. POST.

Note though that distance to *not* is also likely to impact the classifiers' accuracy. Indeed, by definition the structural zones obviously correlate with distance to *not*. For instance, a token at distance 3 to the right of *not* is more likely to be in the licensing scope than a token at distance 20. Hence, to study the impact of the input token's zone, we needed to control for distance to the negation clue.

We thus broke down our classifiers' accuracy on the *not*+NPI test set, not only according to the input token's zone, but also according to its relative position to the negation cue. Table 5 shows an example of *not*+NPI sentence, and the zone and relative position to *not* of each token. The target *not* has position 0, and so do all the PLMs' subword tokens involved in the negation complex, and all preceding modal or auxiliary, to homogenize across PLMs and across contracted/plain negation. By construction, the PRE and PRE-IN zones correspond to negative positions, whereas IN and POST correspond to positive ones.

The break-down by position for ROBERTA-large is shown in Figure 1 (results for other models are in appendix figure 4). Two effects can be observed, for all the 4 PLMs: firstly, there is a general decrease of the accuracy as moving away from *not*, for the four zones. This contrasts with the findings of Klafka and Ettinger (2020), who did not observe a distance effect in their experiments, when probing whether the contextual representation of e.g. a direct object encodes e.g. the animacy of the subject. The decrease is more rapid before *not* than after it, which remains to be explained. It might come from the negation scope being shorter before *not* than after it.

Secondly, when looking at fixed relative distances, there is a slight but consistent effect at almost all positions that the accuracy is higher when the input token is in the negation scope (either PRE-IN or IN), than when it is outside (PRE and POST) (the differences are statistically significant at $p < 0.001$, cf. Appendix B). This tendency is more marked for the PRE *vs.* PRE-IN distinction than for the POST *vs.* IN distinction.

This observation can be summarized by computing the average **accuracy gap**, namely the accuracy differences averaged across positions (the average of the purple minus pink bars, and of blue minus green bars in Figure 3), which provide an average difference when a token is within or outside the negation scope. The average accuracy gaps for the four tested models are given in Table 6. It confirms that input embeddings of tokens inside the negation scope do allow for a slightly better prediction of the presence of *not* than those outside the scope. Note that the average difference is stable across models, whose size does not seem to matter. It shows that the strength of the encoding of *not* in contextual representations matches the linguistic notion of negation scope.

[9]COCA sentences are tokenized and tagged. We detokenized them before BERT/ROBERTA tokenization, in order to get closer to a standard input.

BERT tokens	I	see	I	don	'	t	know	anyone	here	,	I	must	leave	.
Zones	PRE	PRE	PRE-IN	not	not	not	IN	IN	IN	IN	POST	POST	POST	POST
Distance	-3	-2	-1	0	0	0	1	2	3	4	5	6	7	8

ROBERTA tokens	I	see	I	don'	t	know	anyone	here	,	I	must	leave	.
Zones	PRE	PRE	PRE-IN	not	not	IN	IN	IN	IN	POST	POST	POST	POST
Distance	-3	-2	-1	0	0	1	2	3	4	5	6	7	8

Table 5: Example sentence from the *not*+NPI test set: structural zones and relative positions to *not*. Any auxiliary or modal preceding the target *not* has position 0 too, to homogenize contracted and plain negation, and BERT versus ROBERTA's tokenization.

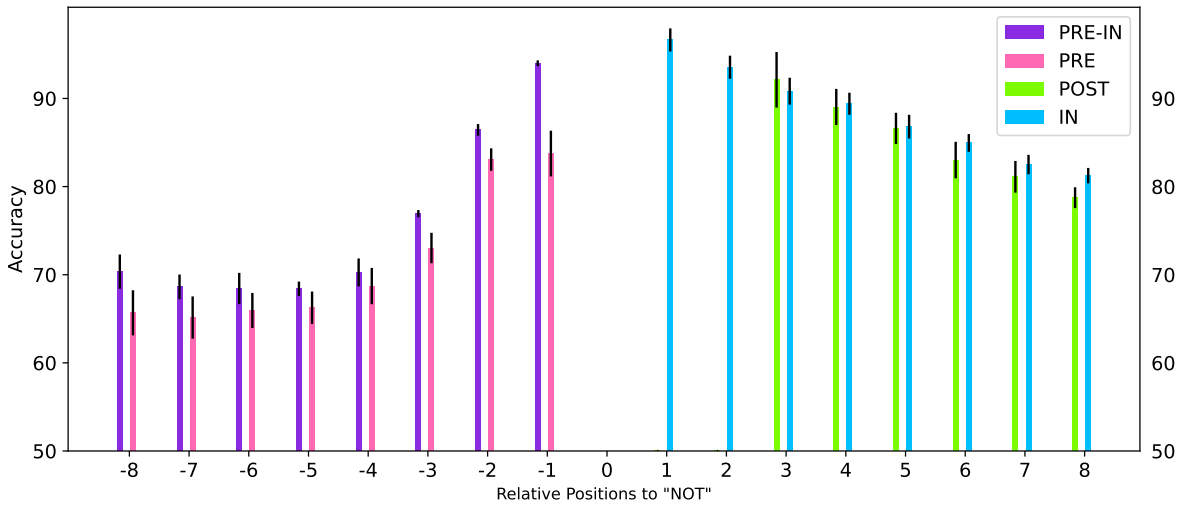


Figure 1: Accuracy of the ROBERTA-large-neg-classifier (average on 3 runs) on the *not*+NPI test set, broken down by zone (colors of the bars) and by relative position to *not* (horizontal axis). Further distances are omitted for clarity. No licensing scope contains less than 2 tokens, hence positions 1 and 2 are always in the IN zone. The bar differences at each position and run are statistically significant at $p < 0.001$ (cf. Appendix B). Figures for the other 3 models are provided in appendix figure 4.

$BERT_b$	$BERT_l$	ROB_b	ROB_l
3.0 (0.6)	3.5 (0.2)	2.6 (0.2)	2.6 (1.3)

Table 6: Accuracy gaps for the neg-classifiers on the *not*+NPI test set, for each tested PLM, averaged over 14 relative positions and 3 runs (stdev within brackets).

We also observed that the biggest difference is at position -1, which mostly corresponds to a contrast between a finite *vs.* non-finite negated verb (neg-patterns 1/2/3 *vs.* neg-pattern 5 in Table 1), which seems well reflected in PLMs' embeddings.

3.2 Probing for the licensing scope

We then focused on whether this encoding of *not* can actually be **mobilized** to capture the licensing of a NPI. We built classifiers (hereafter the ***m*-pol-classifiers**[10], *m* referring to the PLM), taking an input contextual embedding, and predicting as target information the polarity of a masked position, originally filled with a positive or negative PI. Importantly, the input embedding in the training set is randomly chosen in the sentence, and can correspond to a position with no a priori linguistic knowledge about the polarity of the PI (Figure 2).

We train on sentences originally having either a PPI or a NPI, which we mask before running each studied PLM. More precisely, in each COCA subcorpus (each genre), and for each of the 6 NPI/PPI pairs listed by Jumelet and Hupkes (2018)[11], we randomly took at most 2,000 sentences containing

[10] Full details for all classifiers are provided in Appendix A.

[11] *(any/some)(𝕧/where/one/body/thing/time)*

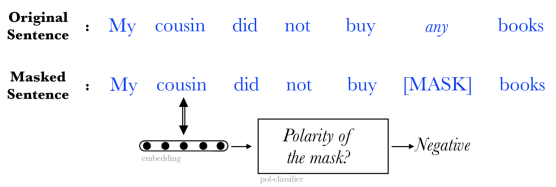


Figure 2: Illustration of the training of the pol-classifiers.

the NPI, and the same amount of sentences containing the corresponding PPI[12]. In each of these, we masked the PI, randomly selected one token per sentence to serve as input token (excluding the masked position) and split these into 63,529 examples for training (**pol-train-set**) and 15,883 for testing (**pol-test-set**).

Model	BERT_b	BERT_l	ROB._b	ROB._l
Accur.	64.2	63.7	56.6	68.6

Table 7: Accuracies of the pol-classifiers on the pol-test-set for each PLM (averaged over 3 runs).

Accuracies on the pol-test-set for each PLM are shown in Table 7. While still above chance, we observe that it doesn't exceed 69%, which is quite lower than the accuracies of the neg-classifiers (Table 4). This is not surprising since the task is more difficult. First, as stressed above, some of the training input tokens are independent, from the linguistic point of view, of the PI's polarity. Second, the cues for predicting the polarity are diverse. And third, in numerous contexts, both polarities are indeed possible, even though not equally likely. We did not control the training for this, on purpose not to introduce any additional bias in the data. We can thus interpret the pol-classifier's scores as how likely a given polarity is.

Next, we applied these classifiers on the ***not*+NPI** test set. The objective is to compare the classifiers' accuracy depending on the structural zone the input token belongs to. If PLMs have a notion of licensing scope, then the polarity prediction should be higher when using an input token from the IN zone.

[12]For *any/some(∅/one/thing)*, we took 2×2000 occurrences. For *any/some(body/time/where)*, less occurrences were available in some of the subcorpora. We took as many as possible, but keeping a strict balance between NPI and PPI sentences (between 2×169 and 2×958 depending on the corpus genre and on the NPI/PPI pair).

3.2.1 Results

Once more, we controlled for distance of the input embedding to *not*. The break-down by position and structural zone for ROBERTA-large is provided in Figure 3 (results for other models are in appendix figures 5).

Again, we observe a general accuracy decrease as moving away from *not*, even faster than for the previous experiment. The decrease is more rapid in the PRE-IN zone than in the IN zone (e.g. at distance -4 in PRE-IN, accuracy is less than 70%, whereas it is still above it at distance 8 in the IN zone), which could indicate that the traces of *not* are more *robust* in the licensing scope.

Secondly, as for the previous experiment, for each relative position, **when the input token is in the negation scope (either PRE-IN or IN), the accuracy is higher than when it is outside (PRE and POST)**. Even though we cannot exclude that the relatively high overall accuracies may be explained by the classifier catching some regularities of the sentences containing a NPI rather than a PPI (independently of the presence of *not*), it remains that for the *not*+NPI sentences, accuracy is higher when the input token is in the negation scope than outside it. Moreover, this trend is much more marked than for the previous experiment.

Thirdly, the amplitude of this observation depends on the model. We provide the accuracy gaps for each PLM in Table 8. We observe that the trend is marked for ROBERTA-large and BERT-base (gap of 8.7 and 7.4 accuracy points, actually much higher than the accuracy gaps for predicting the presence of *not*), but lower for ROBERTA-base and BERT-large.

BERT_b	BERT_l	ROB_b	ROB_l
7.4 (0.5)	3.1 (0.4)	1.4 (0.2)	8.7 (0.6)

Table 8: Accuracy gaps for the pol-classifiers on the *not*+NPI test set, averaged over 14 relative positions and 3 runs (stdev within brackets).

This leads us to conclude that (i) PLMs do encode structural constraints imposed by *not* (NPI licensing), but to varying degrees across the PLMs we tested, and (ii) that this encoding is stronger in the negation scope than outside it, independently of the distance to *not*. This only partially matches the linguistic expectation that the strongest zone should be the licensing scope rather than the entire negation scope.

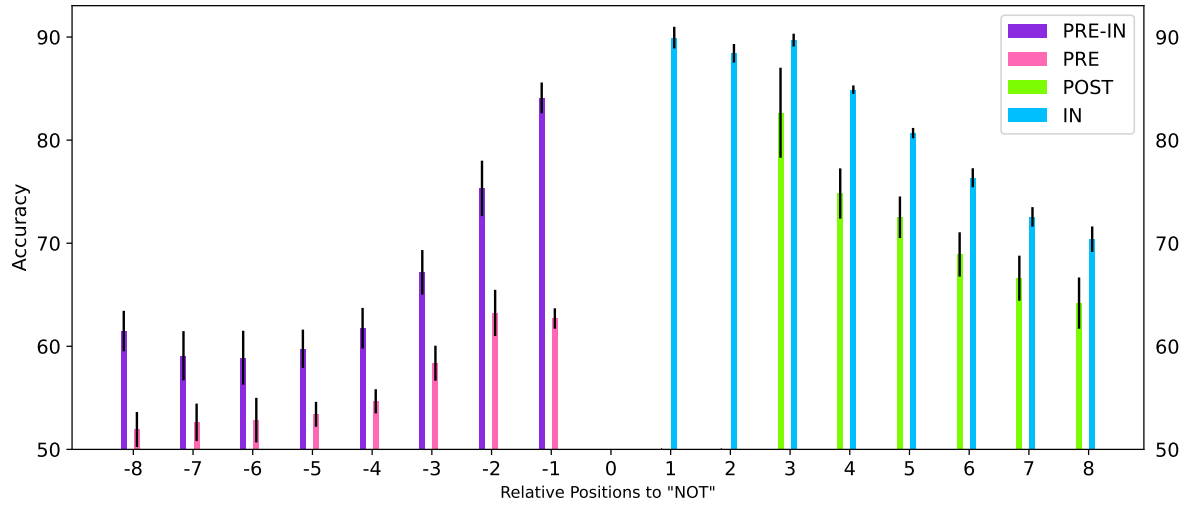


Figure 3: Accuracy of the ROBERTA-large-pol-classifier (average on 3 runs) on the *not*+NPI test set, broken down by zone (colors of the bars) and by relative position to *not* (horizontal axis). Further distances are omitted for clarity. No licensing scope contains less than 2 tokens, hence positions 1 and 2 are always in the IN zone. The bar differences at each position and run are statistically significant at $p < 0.001$ (cf. appendix figures 5).

4 Probing clause boundaries

We have seen that PLMs are able to encode negation scope, however this notion of scope often simply corresponds to the notion of syntactic clause. So it might be the case that PLMs are mainly sensitive to clause boundaries and that this sensitivity is the unique/main source of PLMs ability to encode negation scope. In this section we report a number of experiments designed to assess PLMs ability to encode clause boundaries in general.

We chose to use the same setting as the one we used with the neg-classifiers (section 3.1.1). Instead of using *not* as a target token, we chose various tokens with a similar number of occurrences, but other POSs: *often*, *big*, *house*, *wrote*. We trained classifiers to predict whether the target token is in the neighborhood of the input token. This time, the objective is to compare these classifiers' accuracies depending on whether the input token is or isn't in the same clause as the target token (instead of whether the input token is within or outside the negation scope). And just as we did for the neg-classifiers, we will control for distance to the target token by breaking down the accuracies according to the distance between the target and the input tokens.

4.1 Training the classifiers with alternative target tokens

To train such classifiers, we repeated the same protocol as for the neg-classifiers: for each target word *often*, *big*, *house*, *wrote*, we randomly selected a balanced number of sentences containing and not containing it, and we randomly picked an input token within each sentence, independently of the presence of the target token, and in case of presence, independently of the clause boundary of the target token. We then split the examples into training (25.5k) and test sets (6.5k). We restricted ourselves to a single PLM, ROBERTA-large. The performances on the training and test sets are provided at Table 9. We note that performance is comparable for all the four target tokens, and comparable to that of the neg-classifiers (cf. Table 4, 76.6 for ROBERTA$_l$): the negation clue *not* is not particularly better encoded in contextual embeddings compared to other open-class target words.

Target token	*house*	*often*	*big*	*wrote*
Accur.	79.1	77.1	75.2	81.2

Table 9: Accuracy of the classifiers on test-sets, for the four alternative target tokens, when using ROBERTA-large embeddings (average on 3 runs).

Target token	In	Out	Accuracy gap
house	83.7	79.0	4.7
often	82.0	76.5	5.5
big	80.5	79.4	1.1
wrote	85.7	82.4	3.3

Table 10: Average accuracy when the input token is within a window of 8 tokens before and 8 tokens after the target token, broken down according to whether the input token is (*In*) or isn't (*Out*) in the same clause as the target token, and accuracy gap (*In* minus *Out*). The results are computed the study-test-set of each target word, using the classifiers trained on ROBERTA-large embeddings.

4.2 Studying results when input tokens are within or outside the same clause

In order to study whether PLMs do encode the notion of syntactic clause, we compared the classifiers' performance when the input token is or isn't within the same clause as the target token. For each target word, we built a **study-test-set** of 40,000 COCA sentences containing it. We parsed these sentences, and annotated each of their tokens (1) according to their distance to the target token, and (2) as belonging or not the the same clause as the target token.[13]

As in section 3.1.2, we now define accuracy gaps as the average difference between a classifier accuracy on input tokens that are within the same clause as the target token, minus the accuracy on input tokens from outside the clause. Table 10 shows the average accuracy gaps, for input tokens at distance at most 8 from the target token.

The results show that for the 4 tested target words, predicting the presence of the target token is better achieved using an input token from the same clause than from outside the clause. Interestingly, the gaps are higher when the target token is a noun, verb or adverb, and less pronounced for the adjectival target token. Strikingly, except for the adjective *big*, the observed accuracy gaps are even bigger than that obtained using *not* as target token (cf. 2.6 for ROBERTA$_l$ in Table 6).[14] This tends to indicate that the encoding of the negation scope observed in section 3.1 stems from a more general encoding clause boundaries.

Moreover, breaking-down the results by relative position to the target token (cf. figures 6 in Appendix), shows that the distance to the target token remains by far the most impactful factor.

5 Conclusion

In this paper, we studied the way negation and its scope are encoded in contextual representations of PLMs and to what extent this encoding is used to model NPI licensing.

We trained classifiers to predict the presence of *not* in a sentence given the contextual representation of a random input token. We also trained classifiers to predict the polarity of a masked polar item given the contextual representation of a random token. A test set of sentences was designed with *not* licensing an NPI, inside which we identified the negation scope , and the licensing scope.

For these sentences, we found that the contextual embeddings of tokens within the scope of a negation allow a better prediction of the presence of *not*. These embedding also allow a better prediction of the (negative) polarity of a masked PI. These results hold even when controlling for the distance to *not*. The amplitude of this trend though varies across the four PLMs we tested.

While this tends to indicate that PLMs do encode the notion of negation scope in English, and are able to further use it to capture a syntactic phenomena that depends on the presence of *not* (namely the licensing of a negative polarity item), further experiments tend to show that what is captured is the more general notion of clause boundary. Indeed, negation scope is closely related and often amounts to negation scope. Using alternative target tokens with varied parts-of-speech, we find that classifiers are better able to predict the presence of such target tokens when the input token is within the same syntactic clause than when it is outside from it. These results lead us to conclude that knowledge of the negation scope might simply be a special case of knowledge of clause boundaries. Moreover, distance to the target token is way stronger a factor than the "being in the same clause" factor. We leave for further work the study of other factors, such as the POS of the input token, as well as the study of the differences in amplitudes observed between the PLMs we tested.

[13] We identified the clause of the target token as the subtree of the head verb of the target token, in the dependency parse.

[14] The gaps are not strictly comparable though, due for our defining the negation scope as a subset of the clause, filtering out sentential conjuncts and sentential parenthetical, cf. section 2.3.

Acknowledgements

We thank the reviewers for their valuable comments. This research was partially funded by the Labex EFL (ANR-10-LABX-0083).

References

Hande Celikkanat, Sami Virpioja, Jörg Tiedemann, and Marianna Apidianaki. 2020. Controlling the Imprint of Passivization and Negation in Contextualized Representations. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 136–148, Online. Association for Computational Linguistics.

Mark Davies. 2015. Corpus of Contemporary American English (COCA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Reto Gubelmann and Siegfried Handschuh. 2022. Context matters: A pragmatic study of PLMs' negation understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4602–4621, Dublin, Ireland. Association for Computational Linguistics.

Vincent Homer. 2020. *Negative Polarity*, pages 1–39. John Wiley & Sons, Ltd.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do Language Models Understand Anything? On the Ability of LSTMs to Understand Negative Polarity Items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Brussels, Belgium. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and Misprimed Probes for Pretrained Language Models: Birds Can Talk, But Cannot Fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Josef Klafka and Allyson Ettinger. 2020. Spying on Your Neighbors: Fine-grained Probing of Contextual Embeddings for Information about Surrounding Words. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4801–4811, Online. Association for Computational Linguistics.

Bingzhi Li, Guillaume Wisniewski, and Benoit Crabbé. 2022. How distributed are distributed representations? an observation on the locality of syntactic information in verb agreement tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–507, Dublin, Ireland. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv.

Roser Morante and Eduardo Blanco. 2012. *SEM 2012 Shared Task: Resolving the Scope and Focus of Negation. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 265–274, Montréal, Canada. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Henriëtte de Swart. 1998. Licensing of negative polarity items under inverse scope. *Lingua*, 105(3-4):175–200.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019. Investigating BERT's Knowledge of Language: Five Analysis Methods with NPIs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Yu Zhang, Houquan Zhou, and Zhenghua Li. 2020. Fast and Accurate Neural CRF Constituency Parsing. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*,

pages 4046–4053, Yokohama, Japan. International Joint Conferences on Artificial Intelligence Organization.

A Hyperparameter tuning for the neg-classifiers and the pol-classifiers

The PLMs' contextual representations were obtained using a GeForce RTX 2080 Ti GPU.

The neg-classifiers, the pol-classifiers and the classifiers used to predict the presence of other taget tokens were trained on a CPU, each training taking about 15 minutes. Then, testing them on the *not*+NPI test set took about 5 minutes.

To tune these classifiers, we performed a grid search with: a number of hidden layers included in [1, 2], number of units in each layer in [20, 50, 100 450, 1000], and the learning rate in [1, 0.1, 0.01, 0.001].

We selected a learning rate of 0.001, 2 hidden layers, with size 450 each, based on the accuracies on the neg-test-set and the pol-test-set. Except when the learning rate equaled 1, all hyperparameter combinations resulted in similar performance (less than 1 point of accuracy, in the results of figure 3).

The code and methodology was developed first using the BERT-base model, and then applied to the other models. Including code and methodology development, we estimate that the experiments reported in this paper correspond to a total of 160 hours of GPU computing.

B Statistical significance test

In this section we detail the test performed to assess the statistical significance of the accuracy differences illustrated in Figures 3 and 5.

For each of the four tested PLMs, and for each of 3 runs of classifier training,

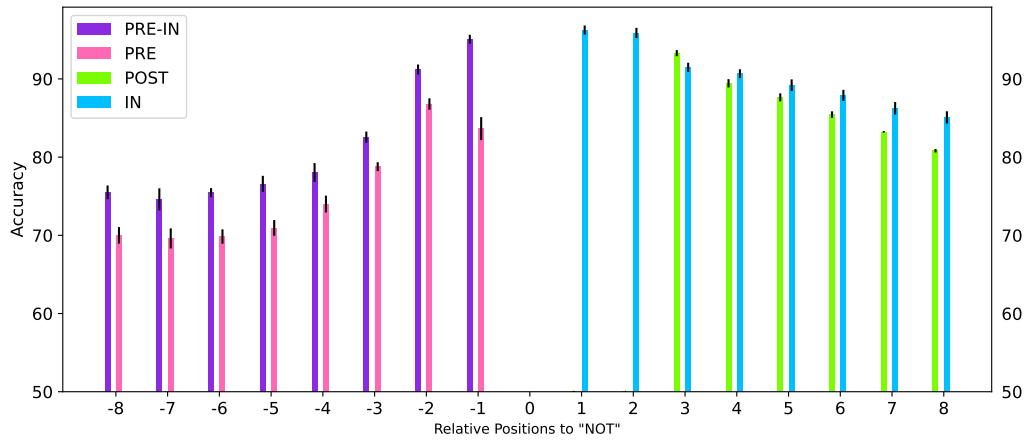
- for each position from -8 to -1 relative to the *not*,
  - we compare the accuracy of the pol-classifier in the PRE-IN zone versus in the PRE zone (i.e. the difference between the purple bar with respect to the pink one).
  - namely, we test the statistical significance of the following positive difference : accuracy for tokens in PRE-IN zone minus accuracy for tokens in the PRE zone.
- for each position from 3 to 8,
  - we test the statistical significance of the following positive difference : accuracy for tokens in IN zone minus accuracy for tokens in the POST zone (i.e. the difference between the blue bar with respect to the green one)

Each test is an approximate Fisher-Pitman permutation test (with 5000 random permutations, performed using the script of Dror et al. (2018), `https://github.com/rtmdrr/testSignificanceNLP.git`), and all the differences listed above result as statistically significant at $p < 0.001$.

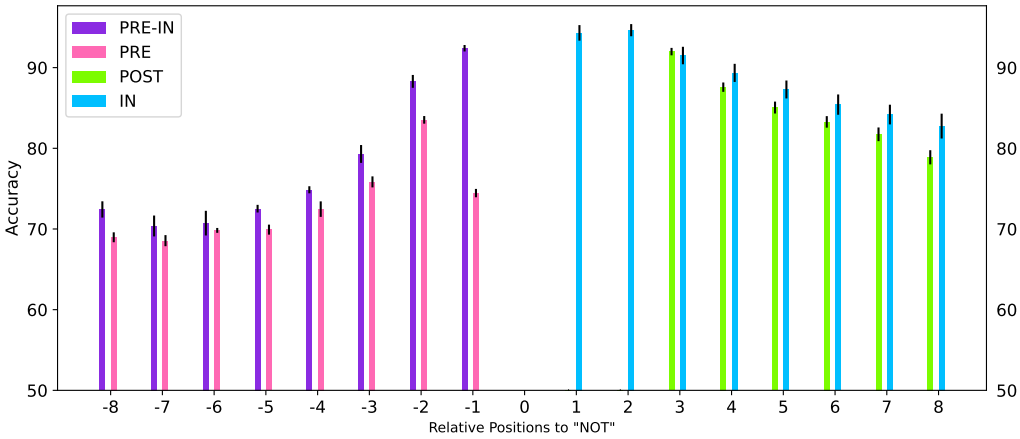
C Supplementary figures

The break-downs by position for the three models not presented in the main text (BERT-base, BERT-large and ROBERTA-base) are provided in Figures 4 (neg-classifiers) and 5 (pol-classifiers).

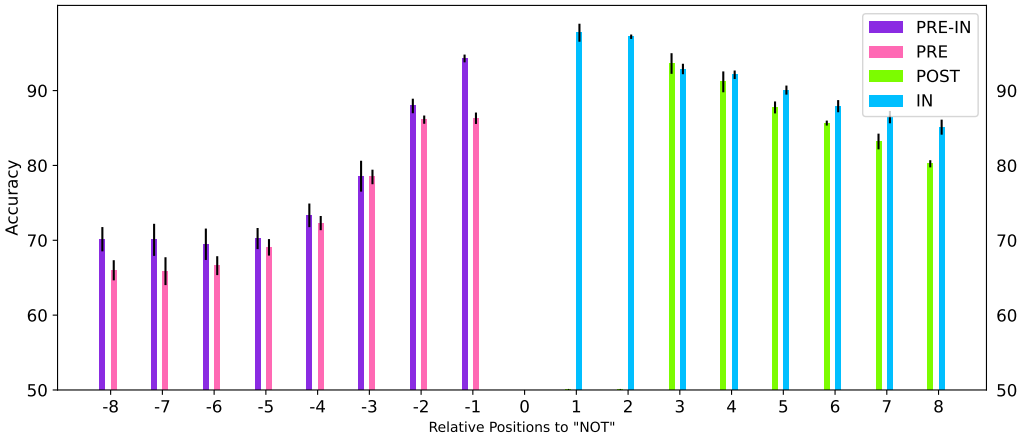
The break-downs by position for other target tokens are provided in Figures 6



(a) BERT-base-cased

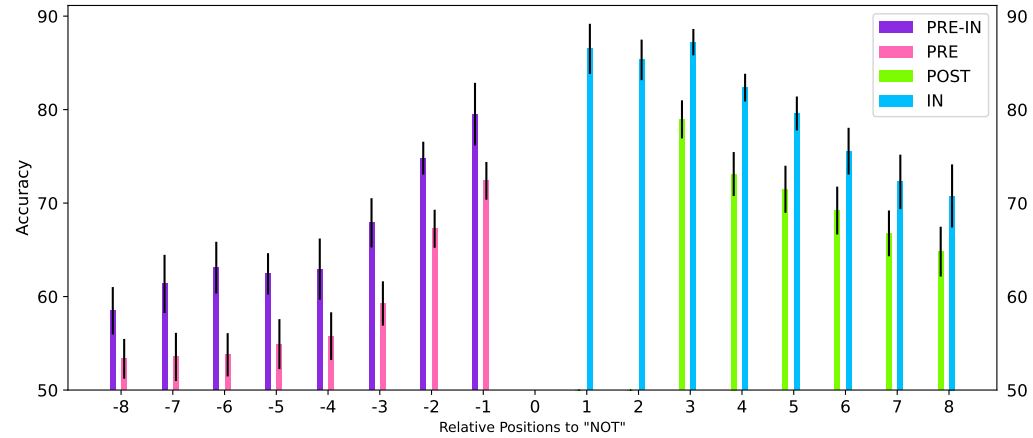


(b) BERT-large-cased

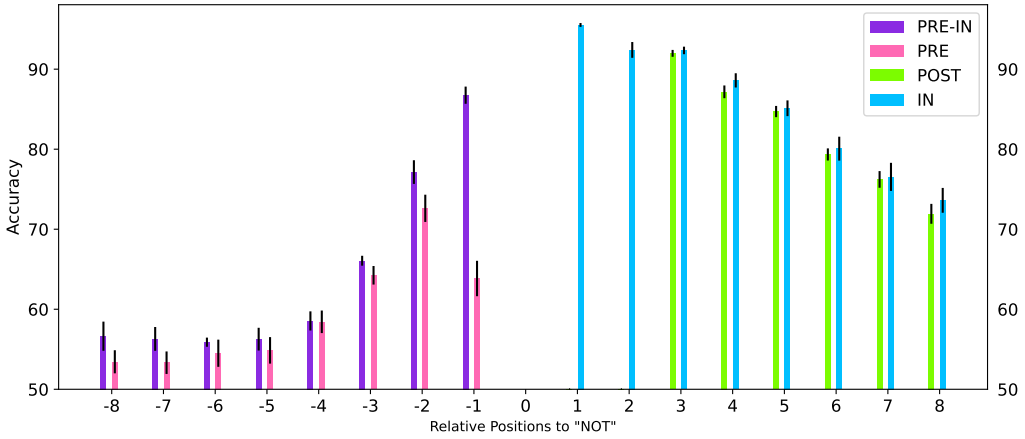


(c) ROBERTA-base

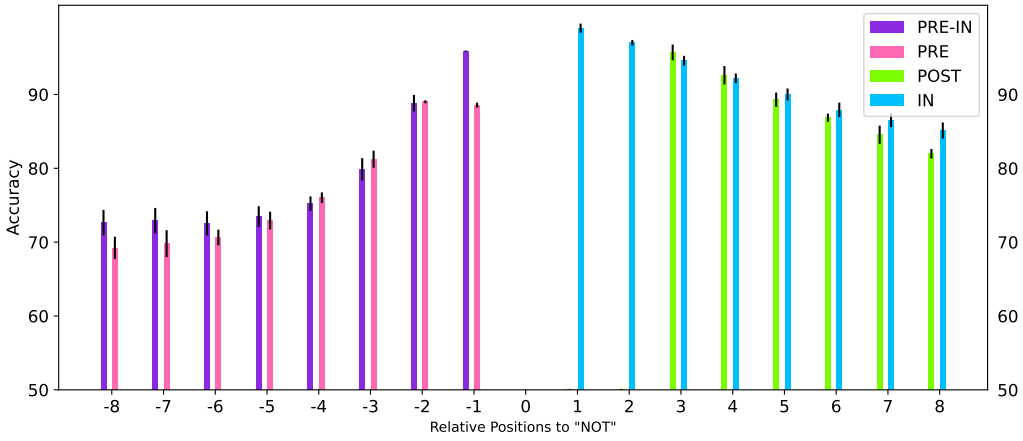
Figure 4: Accuracy (average on 3 runs) of the other neg-classifiers (BERT-base, BERT-large and ROBERTA-base) on the *not*+NPI test set, broken down by zone (colors of the bars) and by relative position to *not* (horizontal axis). Further distances are omitted for clarity. No licensing scope contains less than 2 tokens, hence positions 1 and 2 are always in the IN zone. The bar differences at each position and run are statistically significant at $p < 0.001$ (cf. Appendix B).



(a) BERT-base-cased

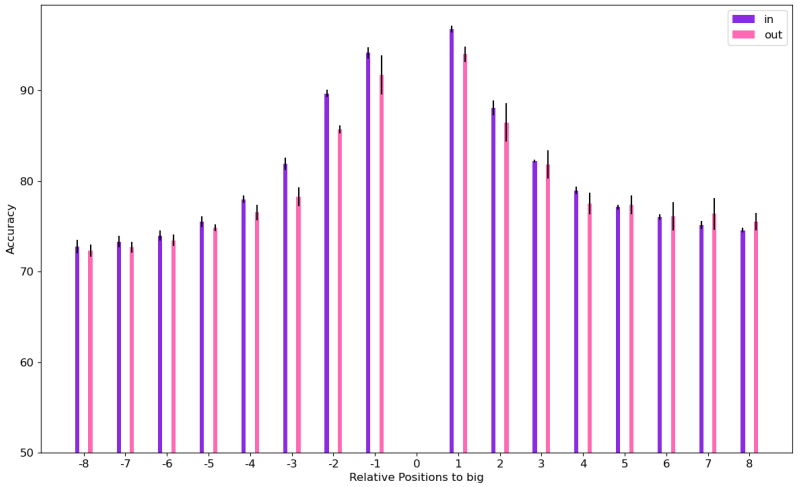


(b) BERT-large-cased

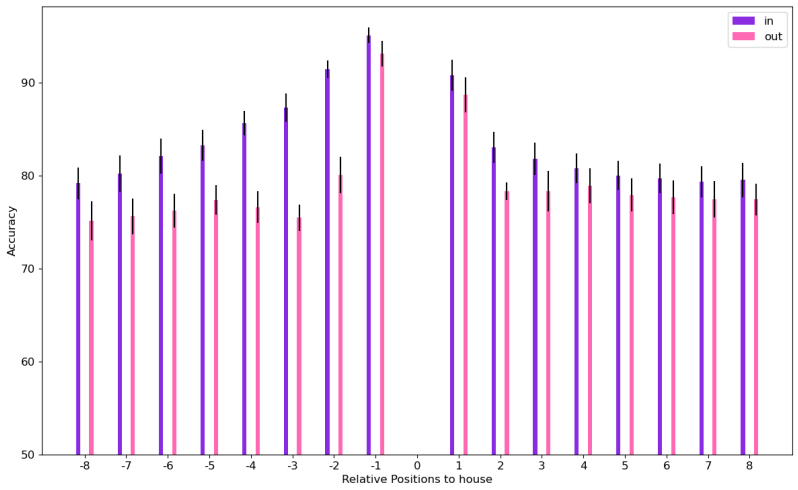


(c) ROBERTA-base

Figure 5: Accuracy (average on 3 runs) of the other pol-classifiers (BERT-base, BERT-large and ROBERTA-base) on the *not*+NPI test set, broken down by zone (colors of the bars) and by relative position to *not* (horizontal axis). Further distances are omitted for clarity. No licensing scope contains less than 2 tokens, hence positions 1 and 2 are always in the IN zone. The bar differences at each position and run are statistically significant at $p < 0.001$ (cf. Appendix B).



(a) big



(b) house

Figure 6: Accuracy (average on 3 runs) on trace identification tasks. The target tokens are *big* and *house*, and the probed embeddings are from a ROBERTA-large LM. Results are broken down by zone (colors of the bars) and by relative position to *not* (horizontal axis). Further distances are omitted for clarity. The bar differences at each position and run are statistically significant at $p < 0.001$ (cf. Appendix B).

Boosting Norwegian Automatic Speech Recognition

Javier de la Rosa versae@nb.no
Rolv-Arild Braaten rolv.braaten@nb.no
Per Egil Kummervold per.kummervold@nb.no
Freddy Wetjen freddy.wetjen@nb.no
Svein Arne Brygfjeld svein.brygfjeld@nb.no

National Library of Norway, Norway

Abstract

In this paper, we present several baselines for automatic speech recognition (ASR) models for the two official written languages in Norway: Bokmål and Nynorsk. We compare the performance of models of varying sizes and pre-training approaches on multiple Norwegian speech datasets. Additionally, we measure the performance of these models against previous state-of-the-art ASR models, as well as on out-of-domain datasets. We improve the state of the art on the Norwegian Parliamentary Speech Corpus (NPSC) from a word error rate (WER) of 17.10% to 7.60%, with models achieving 5.81% for Bokmål and 11.54% for Nynorsk. We also discuss the challenges and potential solutions for further improving ASR models for Norwegian.

1 Introduction

Automatic Speech Recognition (ASR) is the task of converting speech into text. ASR systems are used in a wide range of applications, such as voice assistants, transcription services, and speech-to-text translation. It is also increasingly becoming a tool for research in spoken language as the accuracy of the more recent neural-based models is approaching that of humans for certain metrics. In a study by Amodei et al. (2016), the authors estimated that the word error rate (WER) in human-produced transcriptions on the LibriSpeech benchmark (Panayotov et al., 2015) is roughly 5.83%, while their end-to-end ASR model, DeepSpeech 2, achieved a WER of 5.33% on a clean test set, although it was outperformed by humans on noisy data. Since the introduction of DeepSpeech 2, the field of ASR has progressed even further, with the current leaderboard of the benchmark containing over ten models with a WER below 2%. Despite the high accuracy in resource-rich languages, ASR models are currently unavailable for the vast majority of the world's languages due to the lack of gold annotated data to train such models. Recent advances in unsupervised learning of acoustic models have decreased the need for transcribed speech.

In this paper, we focus on developing and evaluating a new set of baselines ASR models for Norwegian based on the wav2vec 2.0 architecture (Baevski et al., 2020). We make use of existing pre-trained models and combine them with other language resources for the Norwegian languages to further improve the accuracy of the resulting ASR systems. Our models seem to perform notably better than previous work on newly established datasets.

2 Norwegian ASR

The Norwegian language has many spoken dialects, which differ lexically, grammatically, and phonologically. Additionally, there are two official written standards of Norwegian, Bokmål and Nynorsk, which have somewhat different inflection, vocabulary, and spelling. Consequently, high-quality datasets for acoustic modeling of Norwegian require speech data in different dialects and should ideally include transcriptions in both written standards.

Early work on Norwegian speech recognition was mostly focused on very limited vocabularies and numbers, tailored for telephone applications and menu navigation (Svendsen et al., 1989; Paliwal, 1992; Ljøen et al., 1994; Kvale, 1996). Compound words are more frequent in Norwegian than English, but using traditional pronunciation dictionaries seemed sufficient in controlled lexicons. In Norwegian, natural numbers between 20 and 99 can be pronounced differently (e.g. "twenty-four" and "four-and-twenty"), which poses a chal-

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 555–564

lenge for natural number recognition. By the year 2000, and under the umbrella of a few EU-funded projects, research focused mostly on overcoming these limitations and extending the use cases to dates, times, nouns, and the spelling out of words, which yielded several important datasets (e.g., SpeechDat, SpeechDat-II, TABU.0) and technical improvements over a short period of time (Amdal and Ljøen, 1995; Hoge et al., 1997; Kvale and Amdal, 1997; Johansen et al., 1997; Amdal et al., 1999; Martens, 2000). Most approaches were based on hidden Markov models and some relied on Mel Frequency Cepstral Coefficients (MFCC), commonly by using the Hidden Markov Model Toolkit (HTK) (Young and Young, 1993).

However, these approaches were not designed for open-ended recognition and often struggled with out-of-vocabulary words or real conversations. It was not until the introduction of newer datasets in the last decade that systems with reasonable performance started to appear.

2.1 NST

The Nordisk Språkteknologi (NST) dataset is a multi-lingual speech recognition dataset with speech in Swedish, Danish and Norwegian Bokmål, and their corresponding transcriptions. Developed by the now extinct technology company Nordisk Språkteknologi in the late 90s and beginning of the 2000s, the data was manually compiled and mostly validated. It contains telephone conversations, office conversations, read aloud passages, word spellings, and even hesitations. The speaker metadata includes age, gender, region of birth, and regional dialect. The audio quality is generally high, and most recordings have two channels recorded with separate microphones, one placed close to the speaker and one across the room. The dataset comes with training and testing sets. For Norwegian, the training set contains 411.5 hours of speech, while the test contains 115.3 hours. The amount of speech in hours per the regional dialect of the speakers represented in the NST dataset is reported in Table 9 of Appendix C. However, due to its nature as a manuscript-read dataset, the dataset has some limitations, as it only contains planned speech and does not include or contains limited degree of dialectal phenomena which deviate from the Bokmål norm.

2.2 NPSC

In Solberg and Ortiz (2022), the authors present the Norwegian Parliamentary Speech Corpus (NPSC, The National Library of Norway, 2021), an open dataset intended for acoustic modeling of Norwegian unscripted speech. The dataset is developed and distributed by the Language Bank at the National Library of Norway, and consists of approximately 100 hours of recordings of meetings at Stortinget, the Norwegian parliament, in 2017 and 2018. Orthographic transcriptions in Norwegian Bokmål and Norwegian Nynorsk were made. The dataset is public domain and can be used with no restrictions. The dataset is split in training, validation, and test sets (see Table 1).

Solberg and Ortiz trained and tested an ASR system and the results showed that the use of the NPSC dataset improved the recognition performance when compared to the use of only manuscript-read datasets. The authors argue that the NPSC dataset is necessary to fill the gap in the lack of available speech data for Norwegian ASR.

2.3 FLEURS

A very recent addition to the small pool of open datasets suitable for training transformer-based models for ASR comes in the form of a multilingual speech benchmark. The Few-shot Learning Evaluation of Universal Representations of Speech (FLEURS) benchmark (Conneau et al., 2022) is a parallel speech dataset in 102 languages built on top of the FLoRes-101 benchmark for machine translation. FLEURS contains approximately 12 hours of speech per language and can be used for various speech tasks such as automatic speech recognition, speech language identification, translation, and retrieval. The goal of FLEURS is to enable speech technology in more languages and drive research in low-resource speech understanding. The dataset is unique in its coverage of over 100 languages and its suitability for various speech tasks. In their paper, the authors provide baseline results for the different tasks using multilingual pre-trained models, but do not report on single monolingual ones. The almost 11 hours of Norwegian (see Table 2) included in this dataset adhere to Bokmål and represent out of domain speech qualitatively closer to NST than to NPSC.

Language	Train		Validation		Test	
	Hours	**Samples**	**Hours**	**Samples**	**Hours**	**Samples**
Norwegian Bokmål	88.62	44,746	11.70	5,973	11.15	5,527
Norwegian Nynorsk	12.96	6,586	1.61	871	1.33	828
Total	**101.58**	**51,332**	**13.31**	**6,844**	**12.48**	**6,355**

Table 1: Distribution of number of hours and samples for each of the Norwegian written languages in the NPSC dataset.

Train		Validation		Test	
Hours	**Samples**	**Hours**	**Samples**	**Hours**	**Samples**
10.91	3,167	0.58	163	1.25	357

Table 2: Distribution of number of hours and samples for each of the splits in Norwegian subset of the FLEURS dataset.

3 Norwegian wav2vec 2.0

Introduced by Baevski et al. (2020), wav2vec 2.0 is a state-of-the-art self-supervised audio representation learning architecture designed to extract high-quality feature representations from raw audio signals. After pre-training the acoustic model, wav2vec 2.0 models can be used for a wide range of tasks using a regular fine-tuning mechanism. For ASR, these fine-tuned models can be plugged to rather simple n-gram language models that leverage the connectionist temporal classification (CTC) classification loss to further improve recognition.

Wav2vec 2.0 improves upon the original wav2vec architecture by Schneider et al. (2019) in several key ways. First, it uses a transformer-based neural network to predict the audio signal in a context window surrounding a masked center frame. This enables the model to capture long-range dependencies in the audio signal, leading to more accurate feature representations. Second, the model performs multiple prediction tasks simultaneously, including predicting the center frame, predicting the entire context window, and predicting future audio signals. The CTC loss is used to compute the prediction error between the predicted and actual center frame. This multi-task learning approach improves the representational power of the model. Finally, wav2vec 2.0 has a larger number of parameters and a larger training data size, which leads to improved performance on various audio representation learning benchmarks.

In early 2022, we released a series of wav2vec 2.0 models of different sizes. Available for Bokmål in 300 million[1] and 1 billion[2] sizes and for Nynorsk only in 300 million parameters[3], these models were fine-tuned on the NPSC dataset. The 1 billion parameter models were based on the multilingual XLS-R models, and the 300 million parameters models on the Swedish VoxRex model. XLS-R models (Babu et al., 2021) are trained on more than 436,000 hours of publicly available speech recordings. The data used to train the XLS-R models came from a variety of sources, including parliamentary proceedings and audio books, and covered 128 different languages. VoxRex, developed by Malmsten et al. (2022) at National Library of Sweden (KB), is a Swedish acoustic wav2vec 2.0 model trained on the P4-10k corpus which contains 10,000 hours of Swedish local public service radio as well as 1,500 hours of audio books and other speech from KB's collections. The choice of a Swedish acoustic model to fine-tune Norwegian ASR instead of using the same size XLS-R model was motivated by the fact that both languages belong to the North Germanic language family, which all originated from Old Norse, and share many spoken and written features.

4 Methods

In this work, we evaluate these models, referred to as NPSC-Bokmål and NPSC-Nynorsk, and fine-tune new XLS-R 1 billion (1B) parameters and VoxRex 300 million (300M) parameters models

[1] https://huggingface.co/NbAiLab/nb-wav2vec2-300m-bokmaal

[2] https://huggingface.co/NbAiLab/nb-wav2vec2-1b-bokmaal

[3] https://huggingface.co/NbAiLab/nb-wav2vec2-300m-nynorsk

using the same hyperparameters[4]. We train the models on NPSC and ablate on different data supplementing strategies derived from the NST dataset.

The NST dataset was modernized and reorganized by the National Library of Norway, and is now available in a reader-friendly format (Nordisk Språkteknologi, 2020). We omitted the second channel of audio recorded with a distant microphone due to no noticeable differences between the audio recorded with the close microphone. The dataset is representative of the major regions and the language variety spoken in that region, although the representation of the dialectal varieties of the Scandinavian languages in the dataset is debatable (see Appendix C, Table 9). All combinations of NPSC and NST training sets were lowercased, and had removed non-letter characters and accents from characters (aside from the Norwegian 'æøå'). Any samples with an audio clip under half a second are removed. Transcripts containing digits are also removed, as we expect any numbers to be spelled out. NST data containing words spelled out letter by letter were removed, and instructions to stay silent or dictation commands (e.g., comma, period) were replaced with empty strings. For the hesitations in NPSC and NST, most of the runs replace them using triple letters, e.g. `<ee>` becomes `eee`. These models also use the Bokmål translation of the Nynorsk data in NPSC. The resulting models from the different experiments are listed below:

- NST model. Fine-tuned on the NST dataset as described, with no exta modificatons nor additions.
- NST-NPSC model. These models are fine-tuned using the Bokmål and Nynorsk subsets of NPSC plus the NST dataset as described.
- NST-NPSC-Bokmål model. These models are fine-tuned on the Bokmål subset of NPSC plus the translated version of the Nynorsk subset, the NST, and the hesitations subset of NST. These models also replace the hesitations with single letters in the 1 billion parameters models, and the special character $\hat{\text{h}}$ shared between all types of hesitations in the 300 million parameters models since triple letters require a pad character in between.
- NPSC-Nynorsk. Since the NPSC-Nynorsk model was only available as a 300 million parameter model, this model is a 1 billion parameters version fine-tuned on the Nynorsk subset of NPSC plus the translated version of the Bokmål subset.

We trained all models for 40 epochs on a single NVIDIA RTX A6000 GPU with an effective batch size of 24 by accumulating gradients every 2 steps on a batch size of 12. The learning rate was set to $2 \cdot 10^{-5}$, with 2,000 steps of warmup and linear decaying using an Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We used the PyTorch models available in the HuggingFace hub.

After fine-tuning, separate Bokmål and Nynorsk 5-gram Kneser-Ney language model were added where appropriate[5]. Two versions of the NST-NPSC model were also created, one with the Bokmål 5-gram language model, and another one with the Nynorsk language model, as we evaluate the NST-NPSC model on both subsets of NPSC. These language models were created using a combination of the training and validation sets of NPSC plus a few thousand random documents from the Norwegian Colossal Corpus (Kummervold et al., 2021, 2022). We processed a total of 78 million words by lowercasing, normalizing, and filtering out the characters that were outside the 28 Norwegian letters used for fine-tuning. We used the implementation of Kneser-Ney models (Ney et al., 1994) available in the KenLM library (Heafield, 2011). The estimation of the CTC α and β values was done by grid search over $\{0.001, 0.01, 0.1, 0.25, 0.5, 0.75, 1, 1.5, 2, 3\}$ on the validation set of the Bokmål subset of NPSC; we established $\alpha = 0.5$ and $\beta = 0.001$.

5 Results and Discussion

We evaluate the performance of the models grouping their scores by the written language of the test sets in NPSC and NST. We report word error rates as percentages[6]. For comparison purposes, we include the figures obtained in the original NPSC paper by Solberg and Ortiz (2022), as well as the work by Ortiz and Burud (2021) who also briefly evaluated ASR on NPSC. Table 3 shows the WER score of the 300 million and 1 billion parameters

[4]Swedish Wav2vec 2.0 large VoxRex (C) and Multilingual Wav2Vec2-XLSR-53.

[5]`https://huggingface.co/NbAiLab/nb-wav2vec2-kenlm`

[6]For character error rates, please see Appendix B, Tables 6, 7 and 8.

Size	Model	NPSC	NPSC (Bokmål)	NST
300M	*No language model*			
	NPSC-Bokmål	11.76	9.79	21.46
	NST	24.50	22.45	5.52
	NST-NPSC	9.58	8.86	5.44
	NST-NPSC-Bokmål	10.37	8.33	5.49
	5-gram language model			
	NPSC-Bokmål	9.07	7.14	19.19
	NST	19.41	17.33	**4.38**
	NST-NPSC	**7.60**	**6.92**	4.39
	NST-NPSC-Bokmål	10.05	7.96	4.42
1B	*No language model*			
	NPSC-Bokmål	9.49	7.51	17.64
	NST	25.07	22.94	5.08
	NST-NPSC	8.99	7.14	5.25
	NST-NPSC-Bokmål	8.69	6.46	4.93
	5-gram language model			
	NPSC-Bokmål	8.37	6.41	14.94
	NST	21.47	19.36	4.39
	NST-NPSC	8.03	6.15	4.54
	NST-NPSC-Bokmål	**8.02**	**5.81**	**4.30**
	Ortiz and Burud (2021)	20.64		
	Solberg and Ortiz (2022)	17.10		

Table 3: Test sets WER scores of all models fine-tuned on data containing Bokmål. Best scores in **bold** for each size.

models. In both cases, it can be seen that models trained on the Bokmål subset of NPSC perform not too well on the test set of NST. Similary, models trained only on NST underperform on the test set of the Bokmål subset of NPSC. Adding a 5-gram language model yields significant improvements across the board, ranging from a 5 points increase on the worst performing pairs of model and dataset, to a 1 point increase for the best performing pairs. However, the biggest gain in performance is the addition of extra data. The models fine-tuned on combinations of NPSC and NST produce significantly better results. On the whole NPSC, the 300M NST-NPSC model outperform Solberg and Ortiz (2022) by 9.5 points and the previous state of the art NPSC-Bokmål model by 4.16 points. For the other datasets, the 1 billion parameters model NST-NPSC-Bokmål outperformed the rest of models, yielding increases over the NPSC-Bokmål model of 0.6 points on NPSC (Bokmål) subset and of 14.89 points on NST. Interestingly, the performance of the best 300M and 1B models was very close.

An evaluation of the models for each region in the test set of NST can also be found in Appendix C with somewhat similar results and trends. We found that there is virtually no difference in the per region performance of the models, even for the unbalanced (in terms of hours of speech in test set) regions of Oslo and Sør-Vestlandet. It is important to notice that the regions identified in NST do not reflect the diversity of spoken dialects in Norway.

For Nynorsk, as shown in Table 4, our NST-NPSC 300M model with a Nynorsk 5-gram language model attached did not beat the existing NPSC-Nynorsk 300M model. However, our newer NPSC-Nynorsk 1B model outperforms the NPSC-Nynorsk 300M model by 1.14 points.

In order to evaluate the generalization capabilities of our models, we use the Norwegian test set of FLEURS. Transcriptions on FLEURS were normalized as closely as possible to those present in the NST and NPSC, with numbers and times written out in text form. We compare the performance of our models against the Whisper models (Radford et al., 2022), which despite being architecturally different, and being trained in a supervised fashion on almost twice the amount of

Size	Model	NPSC (Nynorsk)
300M	*No language model*	
	NPSC-Nynorsk	16.29
	NST-NPSC	16.52
	5-gram language model	
	NPSC-Nynorsk	**12.68**
	NST-NPSC	14.23
1B	*No language model*	
	NPSC-Nynorsk	13.99
	NST-NPSC	26.99
	5-gram language model	
	NPSC-Nynorsk	**11.54**
	NST-NPSC	25.38

Table 4: Test sets WER scores of all models fine-tuned on data containing Nynorsk. Best scores in **bold** for each size.

hours of XLS-R and with subtitles instead of transcriptions, hold the state of the art on almost every language in FLEURS. However, it is important to notice that their WER scores are calculated on non-normalized text and their parameter counts do not match ours[7]. As shown in Table 5, our best 300 million parameters model more than doubles the performance of Whisper small (244M), with a WER of 9.88 versus 24.20. The 1 billion parameters model NST-NPSC still outperforms Whisper large by 1.53 points, and it is only a negligible 0.37 points from the version 2 of the Whisper large model, that while having 550M fewer parameters than Whisper large.

6 Future Work

Despite the improved performance of our models compared to the other baselines, ASR models for Norwegian still face several challenges. One major challenge is the complex phonetics and morphology of the different dialects, which makes it difficult for models to accurately transcribe the phonemes in the input speech to the correct spelling. Another challenge is the limited availability of high-quality datasets for Norwegian speech, which limits the amount of training data for ASR models.

To address these challenges, one possible to solution is to combine multiple datasets and sources of training data, such as transcribed speech and

[7]Whisper models are able to handle capitalization and punctuation marks.

Size	Model	FLEURS
300M	*No language model*	
	NPSC-Bokmål	18.51
	NST	13.94
	NST-NPSC	12.43
	NST-NPSC-Bokmål	12.51
	5-gram language model	
	NPSC-Bokmål	12.98
	NST	11.27
	NST-NPSC	9.93
	NST-NPSC-Bokmål	**9.88**
	Whisper small (244M)	24.20
1B	*No language model*	
	NPSC-Bokmål	16.26
	NST	13.05
	NST-NPSC	11.17
	NST-NPSC-Bokmål	11.53
	5-gram language model	
	NPSC-Bokmål	13.03
	NST	11.53
	NST-NPSC	**9.87**
	NST-NPSC-Bokmål	10.00
	Whisper large (1.55B)	11.4
	Whisper large-v2 (1.55B)	9.5

Table 5: Test sets WER scores on the Norwegian subset of FLEURS for all models. Best scores in **bold** for each size.

synthetic speech, to increase the amount of pre-training data for ASR models. With enough transcribed speech, even other more data-hungry architectures could be tested, such as Whisper.

Finally, the prospect of training wav2vec 2.0 directly on non-normalized text is an interesting avenue for research, as it would make the models directly usable without having to transform the output of the models to make them more readable.

7 Conclusion

In this paper, we presented several new models for automatic speech recognition of Norwegian. We evaluated these models on several datasets of Norwegian speech and compared their performance to previous work, outperforming the previous state of the art. Given that we used almost the same settings than the wav2vec 2.0 models released last year, with the addition of extra training time and data there are some interesting findings. First, adding over 400 hours of extra planned speech to the semi-improvised speech part of NPSC, performance does not plummet, but actually increases

from 6.41 to 5.81 WER for Bokmål in the 1B settings. The 300M model seems more sensitive in this regard and the WER decreases from 7.14 to 7.96 WER. For NST, the trend is exactly the same, although the differences are smaller.

Interestingly, the out of domain performance of the models is also greatly improved by adding the planned speech in NST to NPSC. Models on both sizes increase their WER scores from 12.98 to 9.88 for the 300M model, and from 13.03 to 9.87 for the 1B model.

We are releasing our best performing models and evaluation code for replicability, and hope to contribute to the advance of ASR for Norwegian.

References

Ingunn Amdal, Trym Holter, and Torbjørn Svendsen. 1999. Maximum likelihood pronunciation modelling of Norwegian natural numbers for automatic speech recognition. In *Proc. Norwegian Signal Processing Symposium (NORSIG)*, pages 145–150.

Ingunn Amdal and Harald Ljøen. 1995. TABU.0 - en norsk telefontaledatabase. *Scientific Report*, 40:95.

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, et al. 2021. XLS-R: Self-supervised cross-lingual speech representation learning at scale. *arXiv preprint arXiv:2111.09296*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2022. FLEURS: FEW-Shot Learning Evaluation of Universal Representations of Speech. *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Harald Hoge, Herbert S. Tropf, Richard Winski, Henk van den Heuvel, Reinhold Haeb-Umbach, and Khalid Choukri. 1997. European speech databases for telephone applications. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1771–1774. IEEE.

Finn Tore Johansen, Ingunn Amdal, and Knut Kvale. 1997. The Norwegian part of speechdat: A European speech database for creation of voice driven teleservices. *Proceedings of NORSIG-1997*.

Per Egil Kummervold, Javier De la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. 2021. Operationalizing a national digital library: The case for a Norwegian transformer model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 20–29, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Per Egil Kummervold, Freddy Wetjen, and Javier De la Rosa. 2022. The Norwegian colossal corpus: A text corpus for training large Norwegian language models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3852–3860, Marseille, France. European Language Resources Association.

Knut Kvale. 1996. Norwegian numerals: A challenge to automatic speech recognition. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, volume 4, pages 2028–2031. IEEE.

Knut Kvale and Ingunn Amdal. 1997. Improved automatic recognition of Norwegian natural numbers by incorporating phonetic knowledge. *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 3:1763–1766 vol.3.

Harald Ljøen, Ingunn Amdal, and Finn Tore Johansen. 1994. Norwegian speech recognition for telephone applications. In *Proc. Norsig*, volume 94, pages 121–125.

Martin Malmsten, Chris Haffenden, and Love Börjeson. 2022. Hearing voices at the National Library–a speech corpus and acoustic model for the Swedish language. *arXiv preprint arXiv:2205.03026*.

Jean-Pierre Martens. 2000. Final report of COST action 249: Continuous speech recognition over the telephone. Technical report, Electronics & Information Systems, Ghent University.

Hermann Ney, Ute Essen, and Reinhard Kneser. 1994. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech & Language*, 8(1):1–38.

Nordisk Språkteknologi. 2020. NST Norwegian ASR Database (16 kHz) – Reorganized.

Pablo Ortiz and Simen Burud. 2021. BERT attends the conversation: Improving low-resource conversational ASR. *arXiv preprint arXiv:2110.02267*.

Kuldip K. Paliwal. 1992. On the use of line spectral frequency parameters for speech recognition. *Digital signal processing*, 2(2):80–87.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an ASR corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *Proc. Interspeech 2019*, pages 3465–3469.

Per Erik Solberg and Pablo Ortiz. 2022. The Norwegian parliamentary speech corpus. *arXiv preprint arXiv:2201.10881*.

Torbjørn Svendsen, Kuldip K. Paliwal, Erik Harborg, and PO Husoy. 1989. An improved sub-word based speech recognizer. In *International Conference on Acoustics, Speech, and Signal Processing,*, pages 108–111. IEEE.

The National Library of Norway. 2021. Norwegian Parliamentary Speech Corpus.

Steve J. Young and S. Young. 1993. *The HTK hidden Markov model toolkit: Design and philosophy*. University of Cambridge, Department of Engineering Cambridge.

A Availability

The best performing models and the code use to train and evaluate them are released with a permissive license in a model hub:

- NST-NPSC 300M model as `nb-wav2vec2-300m-bokmaal-v2`.
- NST-NPSC-Bokmål 1B model as `nb-wav2vec2-1b-bokmaal-v2`.
- NPSC-Nynorsk 300M model as `nb-wav2vec2-300m-nynorsk`.
- NPSC-Nynorsk 1B model as `nb-wav2vec2-1b-nynorsk`.

The results raw data is also available in a code repository to replicate all tables and figures in this work at `nb-wav2vec2`.

B Character Error Rates (CER)

Size	Model	NPSC (Nynorsk)
300M	*No language model*	
	NPSC-Nynorsk	4.91
	NST-NPSC	5.03
	5-gram language model	
	NPSC-Nynorsk	**4.38**
	NST-NPSC	4.80
1B	*No language model*	
	NPSC-Nynorsk	4.52
	NST-NPSC	7.33
	5-gram language model	
	NPSC-Nynorsk	**4.12**
	NST-NPSC	7.07

Table 6: Test sets CER scores of all models fine-tuned on data containing Nynorsk. Best scores in **bold** for each size.

Size	Model	FLEURS
300M	*No language model*	
	NPSC-Bokmål	4.96
	NST	3.96
	NST-NPSC	3.48
	NST-NPSC-Bokmål	3.46
	5-gram language model	
	NPSC-Bokmål	3.83
	NST	3.46
	NST-NPSC	2.92
	NST-NPSC-Bokmål	**2.89**
1B	*No language model*	
	NPSC-Bokmål	4.42
	NST	3.88
	NST-NPSC	3.13
	NST-NPSC-Bokmål	3.24
	5-gram language model	
	NPSC-Bokmål	3.73
	NST	3.58
	NST-NPSC	**2.89**
	NST-NPSC-Bokmål	2.91

Table 7: Test sets CER scores on the Norwegian subset of FLEURS for all models. Best scores in **bold** for each size.

Size	Model	NPSC	NPSC (Bokmål)	NST
300M	*No language model*			
	NPSC-Bokmål	3.63	3.13	5.05
	NST	8.84	8.23	1.75
	NST-NPSC	3.08	2.87	1.70
	NST-NPSC-Bokmål	3.07	2.55	1.76
	5-gram language model			
	NPSC-Bokmål	3.24	2.74	4.59
	NST	8.17	7.53	**1.55**
	NST-NPSC	**2.83**	**2.62**	1.52
	NST-NPSC-Bokmål	3.31	2.75	1.56
1B	*No language model*			
	NPSC-Bokmål	3.17	2.67	4.23
	NST	9.32	8.65	1.63
	NST-NPSC	2.99	2.54	1.65
	NST-NPSC-Bokmål	2.71	2.06	1.64
	5-gram language model			
	NPSC-Bokmål	3.01	2.51	3.69
	NST	8.75	8.09	1.52
	NST-NPSC	2.85	2.39	1.53
	NST-NPSC-Bokmål	**2.62**	**1.98**	**1.53**

Table 8: Test sets CER scores of all models fine-tuned on data containing Bokmål. Best scores in **bold** for each size.

C NST regions

Region	Train		Test	
	Hours	**Samples**	**Hours**	**Samples**
Oslo-området	53.2	38,688	25.3	17,729
Ytre Oslofjord	48.0	34,008	7.3	4,935
Bergen og Ytre Vestland	45.7	31,824	8.3	5,922
Sør-Vestlandet	42.2	29,328	10.3	6,909
Trøndelag	38.4	27,456	9.3	5,922
Sørlandet	36.9	26,600	9.0	5,922
Voss og omland	33.6	22,776	9.4	5,922
Troms	30.5	19,344	9.6	4,935
Nordland	28.0	20,591	8.8	5,922
Total	**411.5**	**289,934**	**115.3**	**75,965**

Table 9: Distribution of number of hours and speakers for each of the dialect regions (region of youth) of the Norwegian subset of the NST dataset.

Size	Region	NPSC-Bokmål	NPSC-Nynorsk	NST	NST-NPSC	NST-NPSC-Bokmål
300M	*No language model*					
	Bergen og Ytre Vestland	26.14 / 6.24	45.57 / 11.34	6.18 / 1.77	5.92 / 1.75	5.92 / 1.75
	Hedmark og Oppland	19.22 / 4.08	42.36 / 10.31	4.71 / 1.12	4.56 / 1.06	4.56 / 1.14
	Nordland	21.92 / 4.67	42.62 / 10.03	5.21 / 1.27	5.00 / 1.20	4.99 / 1.25
	Oslo-området	20.21 / 5.90	42.50 / 11.87	6.67 / 3.29	6.60 / 3.21	6.65 / 3.26
	Sunnmøre	22.72 / 5.00	41.56 / 9.64	5.02 / 1.16	5.04 / 1.15	5.10 / 1.21
	Sør-Vestlandet	24.55 / 5.78	45.44 / 11.53	6.23 / 1.57	6.13 / 1.53	6.25 / 1.60
	Sørlandet	21.99 / 4.77	44.04 / 10.52	5.52 / 1.34	5.45 / 1.31	5.48 / 1.37
	Troms	21.66 / 4.35	42.56 / 9.71	4.21 / 0.97	4.25 / 0.95	4.28 / 1.01
	Trøndelag	18.28 / 3.77	40.26 / 9.54	4.32 / 1.02	4.27 / 1.00	4.43 / 1.07
	Voss og omland	20.22 / 4.32	38.84 / 8.86	4.10 / 0.98	4.21 / 0.98	4.24 / 1.04
	Ytre Oslofjord	21.08 / 4.69	44.45 / 11.36	6.04 / 1.54	5.87 / 1.46	5.94 / 1.50
	5-gram language model					
	Bergen og Ytre Vestland	22.75 / 5.58	42.64 / 10.81	4.91 / 1.52	4.83 / 1.54	4.70 / 1.55
	Hedmark og Oppland	17.69 / 3.75	39.31 / 9.81	3.58 / 0.94	3.48 / 0.88	3.58 / 0.96
	Nordland	19.59 / 4.20	39.76 / 9.58	3.89 / 1.04	3.89 / 1.00	3.91 / 1.04
	Oslo-området	18.39 / 5.53	39.48 / 11.36	5.70 / 3.10	5.66 / 3.04	5.67 / 3.07
	Sunnmøre	19.74 / 4.39	38.94 / 9.25	3.90 / 0.98	4.04 / 0.99	4.03 / 1.04
	Sør-Vestlandet	21.44 / 5.17	42.45 / 10.97	4.74 / 1.31	4.84 / 1.31	4.87 / 1.36
	Sørlandet	19.64 / 4.30	41.34 / 10.03	4.24 / 1.12	4.29 / 1.10	4.29 / 1.14
	Troms	19.10 / 3.87	39.90 / 9.31	3.17 / 0.79	3.26 / 0.78	3.35 / 0.85
	Trøndelag	16.78 / 3.48	36.68 / 8.94	3.34 / 0.85	3.38 / 0.84	3.55 / 0.92
	Voss og omland	18.56 / 3.97	36.18 / 8.50	3.32 / 0.85	3.36 / 0.83	3.37 / 0.89
	Ytre Oslofjord	18.53 / 4.14	40.97 / 10.75	4.51 / 1.26	4.53 / 1.21	4.57 / 1.24
1B	*No language model*					
	Bergen og Ytre Vestland	21.88 / 5.41	43.60 / 11.38	5.47 / 1.61	5.85 / 1.73	5.17 / 1.61
	Hedmark og Oppland	14.48 / 3.06	39.23 / 9.89	4.29 / 1.02	4.29 / 0.99	4.15 / 1.04
	Nordland	17.78 / 3.79	40.67 / 9.97	4.44 / 1.08	4.68 / 1.11	4.39 / 1.10
	Oslo-området	16.70 / 5.10	40.45 / 11.58	6.27 / 3.15	6.30 / 3.12	6.14 / 3.16
	Sunnmøre	20.41 / 4.57	39.73 / 9.67	4.59 / 1.07	5.19 / 1.19	4.51 / 1.09
	Sør-Vestlandet	20.84 / 5.02	43.82 / 11.54	6.23 / 1.55	6.19 / 1.55	6.03 / 1.54
	Sørlandet	17.69 / 3.72	41.63 / 10.26	5.23 / 1.29	5.30 / 1.27	4.92 / 1.24
	Troms	17.22 / 3.42	40.30 / 9.44	3.75 / 0.86	3.92 / 0.87	3.47 / 0.83
	Trøndelag	14.16 / 3.01	37.28 / 9.23	3.80 / 0.89	4.07 / 0.91	3.65 / 0.90
	Voss og omland	16.50 / 3.52	36.05 / 8.62	3.72 / 0.90	4.04 / 0.93	3.78 / 0.93
	Ytre Oslofjord	17.69 / 3.83	43.30 / 11.26	5.32 / 1.38	5.47 / 1.36	5.25 / 1.39
	5-gram language model					
	Bergen og Ytre Vestland	18.34 / 4.67	41.20 / 10.92	4.69 / 1.49	5.03 / 1.58	4.54 / 1.50
	Hedmark og Oppland	12.37 / 2.64	36.81 / 9.49	3.65 / 0.93	3.61 / 0.88	3.63 / 0.96
	Nordland	14.97 / 3.24	38.22 / 9.55	3.68 / 0.96	3.88 / 0.97	3.70 / 0.99
	Oslo-området	14.53 / 4.68	37.83 / 11.11	5.67 / 3.04	5.71 / 3.01	5.57 / 3.05
	Sunnmøre	16.79 / 3.82	37.81 / 9.36	3.91 / 0.96	4.46 / 1.07	3.88 / 1.00
	Sør-Vestlandet	17.63 / 4.32	41.68 / 11.09	5.30 / 1.41	5.31 / 1.40	5.19 / 1.41
	Sørlandet	14.95 / 3.19	39.27 / 9.86	4.39 / 1.16	4.46 / 1.13	4.16 / 1.11
	Troms	14.54 / 2.95	37.90 / 9.05	3.16 / 0.76	3.27 / 0.77	3.04 / 0.77
	Trøndelag	11.86 / 2.54	34.62 / 8.70	3.27 / 0.80	3.44 / 0.80	3.11 / 0.81
	Voss og omland	13.43 / 2.93	34.14 / 8.36	3.19 / 0.83	3.51 / 0.84	3.23 / 0.85
	Ytre Oslofjord	15.36 / 3.35	40.32 / 10.72	4.42 / 1.22	4.64 / 1.20	4.41 / 1.24

Table 10: Per region test set word and character error rates (WER / CER) of all models fine-tuned on NST.

Length Dependence of Vocabulary Richness

Niklas Zechner
Språkbanken
University of Gothenburg
niklas.zechner@gu.se

Abstract

The relation between the length of a text and the number of unique words is investigated using several Swedish language corpora. We consider a number of existing measures of vocabulary richness, show that they are not length-independent, and try to improve on some of them based on statistical evidence. We also look at the spectrum of values over text lengths, and find that genres have characteristic shapes.

1 Introduction

Measures of lexical richness have several uses, including author identification, other forms of text classification, and estimating how difficult a text is. One of the simplest and most obvious measures of lexical richness is to compare the size of the vocabulary (that is, how many different words) to the size of the text (how many words in total). This can be done in several ways, most straightforwardly as the type-token ratio (henceforth TTR), u/n, where u is the number of unique words (types) and n is the total number of words (tokens). Thus, for the sentence "this example is this example", there are three types and five tokens, so TTR is $u/n = 3/5 = 0.6$.

The obvious problem with TTR is that it changes with the length of the text. As we write a text, the more words we have already written, the more likely it is that the next word will be one that has already been used, so TTR goes down as the text grows longer. Many attempts have been made to transform this measure into something independent of the length of the text, but many of those attempts were made in an age before "big data", or even before computers, and were based on a priori reasoning rather than statistical analysis (Tweedie and Baayen, 1998).

We will start by looking at some of these measures, and test them on a set of corpora to see how they hold up for a wide range of different n. After comparing some of the previous methods, we will briefly look into using the empirical data to come up with a better suggestion. The results give rise to another question: What if instead of aiming for a length-independent measure, we consider *how* the values change with the length? Can that actually tell us new and interesting things?

We find that if we analyse the type count for different sample lengths, we see clear and consistent differences between different types of text. This may be useful for genre classification, or for a more detailed description of the text complexity.

Although these measures are usually applied to specific texts, we here apply them to entire corpora. We will discuss the effects of this after seeing the results.

2 Data

Språkbanken (the Swedish Language Bank) at the University of Gothenburg (spraakbanken.gu.se) has a large collection of text corpora, mainly in Swedish but including several other languages. In this study, we use Swedish texts, focusing on large and homogeneous corpora, listed in the appendix.

We extract the type count u for several different lengths n. Words are case-independent but otherwise counted as written, without lemmatisation. For each n, we divide the corpus in chunks of length n, dropping any overflow at the end, and take the mean value of u for each of these chunks. (In some cases we remove the last value for being an outlier; presumably this is because it is the only value where a large part of the data is dropped due to overflow.) We use a pseudo-logarithmic scale for ease of reading, extracting values for $n = 10, 20, 50, 100, 200, 500, 1000...$ up to the maximum possible for each corpus; the largest go up to 500 million tokens.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 565–573

3 Testing existing measures

First of all, we can test and verify that TTR does go down. Figure 1 shows TTR for 31 corpora.

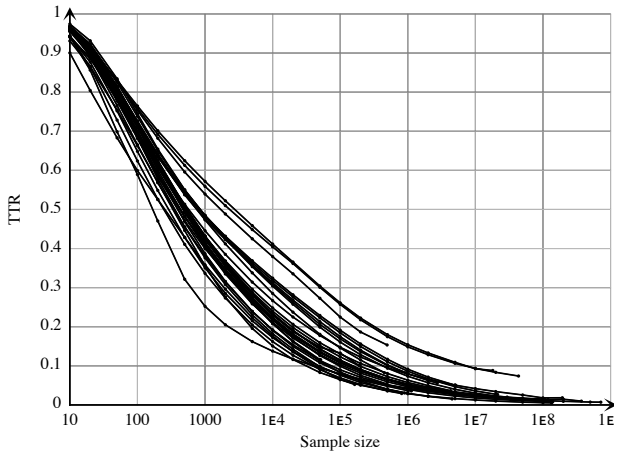


Figure 1: Type-token ratio

It seems likely that, as we compare different-size corpora, effects of size changes might be best described in terms of multiplicative changes rather than additive, so we might try looking at the logarithms of n and u. We see in Figure 2 that the result looks fairly close to a straight line.

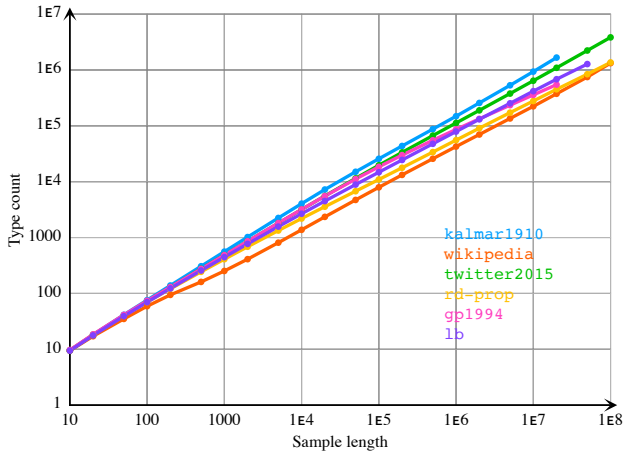


Figure 2: Type count

The first obvious method, then, is to assume that this is indeed a straight line, and use the slope of that line as our presumed length-independent measure of richness, that is, $\log u / \log n$. This was proposed by Herdan (1964). We see in Figure 3 that the measure is decreasing quite steadily for all the texts. The six corpora used here are chosen partly for being large, and partly for having large differences in type count; many other corpora are not nearly as well separated.

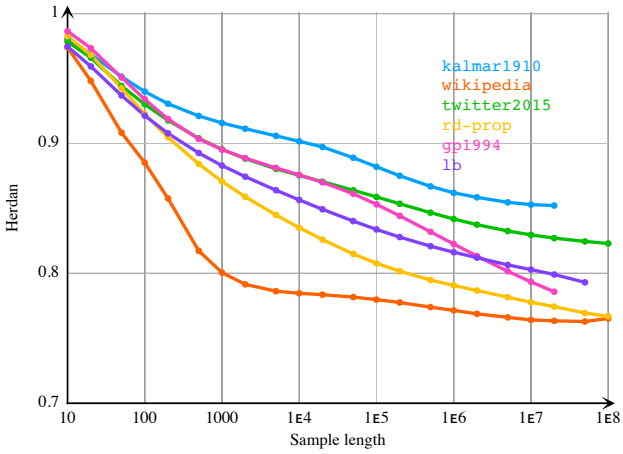


Figure 3: Herdan's measure

Let us pause for a moment and consider what this figure illustrates. The fact that the measure decreases is not in itself a problem; although we are aiming for a near-constant, we should not expect it to be perfect. The amount of variation is also not relevant; we could change that by adding or multiplying by a constant. Regardless of how large the variation is, we would also change the axes of the graph, so a glance at the variation of a single curve in the graph does not tell us whether the measure is near-constant in a relevant sense.

What actually matters is comparing the curves. If the measure is to reliably compare different texts, regardless of the (sample) size for each text, what we need is to have the lines separated insofar as possible. If the lowest point of curve A is higher than the highest point of curve B, then we have successfully determined that A has a higher richness. We should also keep in mind that the first few points of the curve are not as important – we are probably not very interested in measuring richness for very short texts, so although the graphs go all the way from 10, we can mostly ignore values below 1000 or so. We would be content if the measure can separate the lines from that point on.

As we see in Figure 3, this is not quite the case here. This measure works better than TTR, but the curves are still close enough that their ranges overlap. We will compare with a few other measures.

Guiraud (in 1954, as cited by Hultman and Westman (1977)) proposed the measure $u/\sqrt{n}$,

shown in Figure 4. This does not separate the curves particularly well, and does not seem to have any advantage over the previous method.

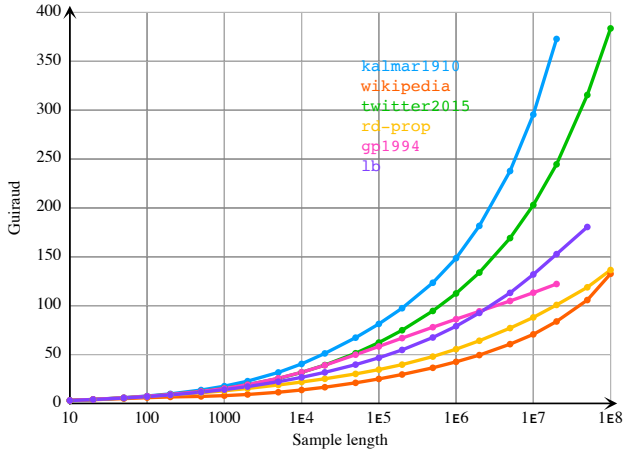


Figure 4: Guiraud's measure

Dugast (1979) built on Herdan by suggesting $\log u / \log \log n$, seen in Figure 5. We find no advantage with this method, and only added conceptual complexity with the double logarithm.

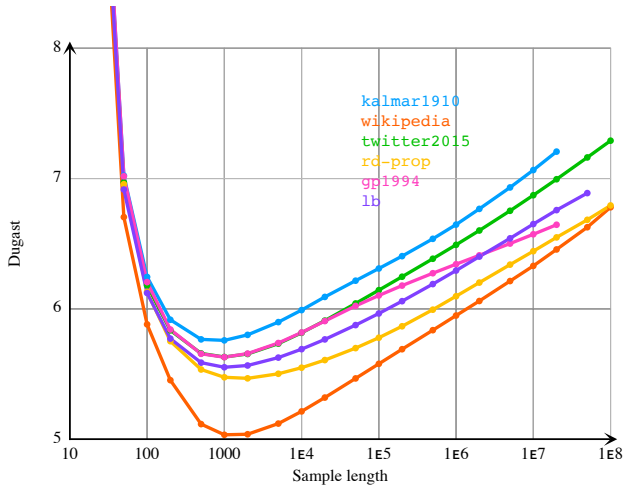


Figure 5: Dugast's measure

Brunet (1978) proposed $n^\wedge(u^{-a})$, where usually $a = 0.172$. This is shown in Figure 6. This too is a fairly conceptually complicated method which shows no sign of improving the results.

Maas (1972) found another approach, with $(\log n - \log u)/(\log n)^2$, see Figure 7. This seems marginally more effective at separating the curves.

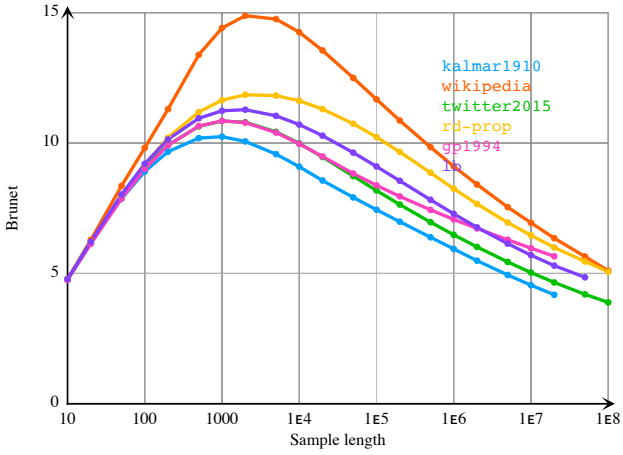


Figure 6: Brunet's measure

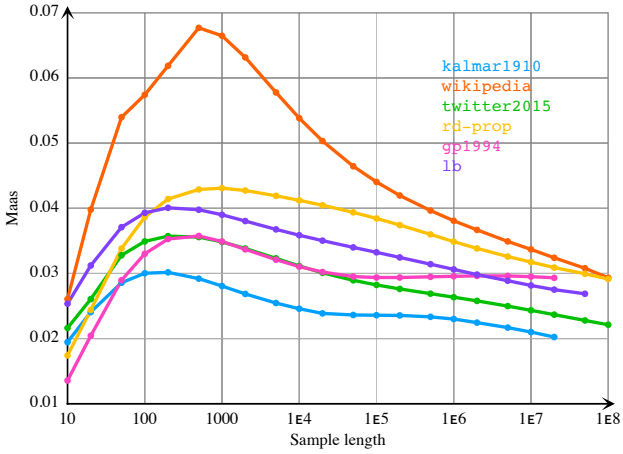


Figure 7: Maas's measure

Hultman and Westman (1977) defined the OVIX measure as

$$\frac{\log n}{\log\left(2 - \frac{\log u}{\log n}\right)}$$

which is seen in Figure 8. This is a measure commonly used in Sweden, including by Språkbanken. As we see, this also does a passable job, but there is a clear rising trend for most curves. This is confirmed by further testing on other corpora.

4 Improving measures

By analysing the way these measures depend on n, we may be able to adjust and improve them. As noted, the fact that the curve of $\log u$ against $\log n$ is close to a line suggests that $(\log u/(\log n)$ may

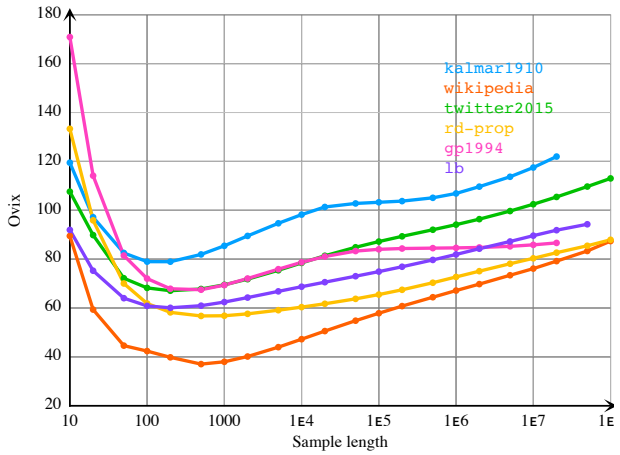


Figure 8: Ovix

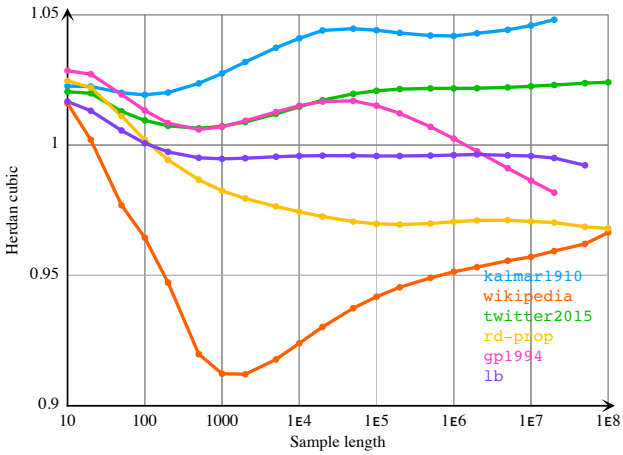


Figure 10: Herdan with cubic fit

be a constant, as per Herdan. But that assumes that the line passes through $(0, 0)$; if the line passes though $(0, m)$ for some m, we should expect that $(\log u - m)/\log n$ is constant. We find that for a subset of the corpora, the best-fitting line gives $m = 0.4$, and we see in Figure 9 that $(\log u - 0.4)/\log n$ does look a lot flatter. As before, we pay less attention to the values where $n < 1000$.

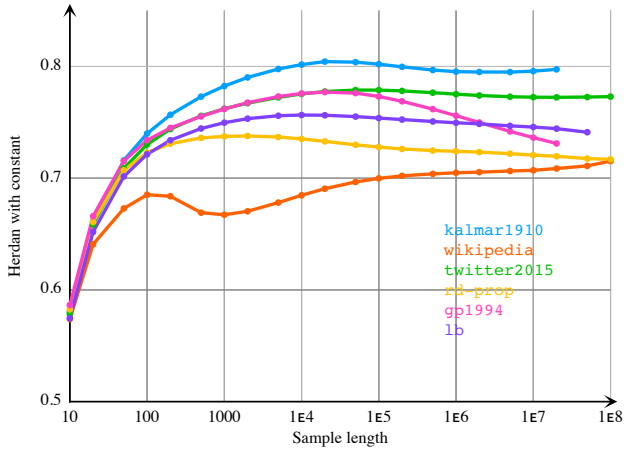


Figure 9: Herdan with constant term

On the other hand, we know that a text with one word certainly also has one unique word, so logically the curve of $\log u$ against $\log n$ *must* pass though $(0, 0)$. Empiricism is all good and well, but if we want results that hold up for other data, perhaps we are better off not violating basic logic. What if instead of a line, we fit the points to a polynomial curve with zero constant term? Trying

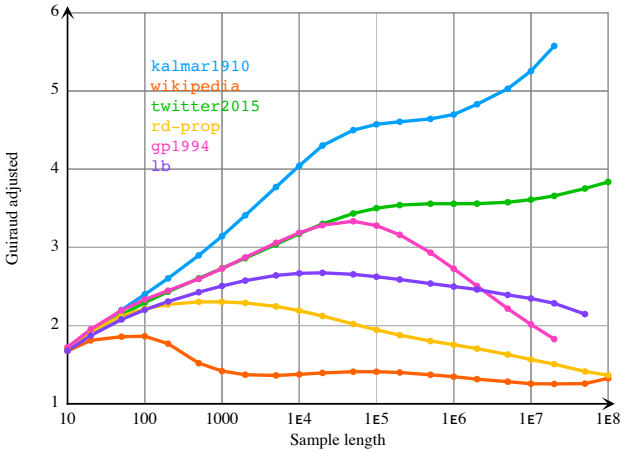


Figure 11: Adjusted Guiraud

second, third and fourth order polynomials suggests that third is a good compromise. We find the best fit for six corpora, take the average for the quadratic and cubic terms, and get the adjusted measure

$$\log u/\log n + 0.044(\log n) - 0.0024(\log n)^2$$

You can see in Figure 10 that this separates the curves considerably better than the pure Herdan measure. From looking at the graph, this is probably the best option we have here, but we should note that the coefficients vary quite a bit between corpora (standard deviations are 0.015 and 0.0017), so this is not universal enough to adopt as some sort of standard measure.

We can also consider the Guiraud approach, and try to adjust it. We notice that while TTR (u/n)

goes steadily down, Guiraud ($u/n^{0.5}$) goes up. Perhaps we can find a middle ground? Figure 11 shows the results for $u/n^{0.75}$, which looks overall much flatter and better separating the curves. This may not be a better result than the previous one, but it does have the advantage of not depending on experimentally determined coefficients.

5 Fixed-length measures

Is there another option, using only the length and the type count? Yes, there is an option which is in principle completely independent of text length: Measure the type count (or equivalently TTR) for a fixed length. One option would be to measure only the first n words of a text, but that could mean that a small part of the text has a large impact, so probably a better method is to cut the text into pieces of length n and take the average, exactly as we have done above.

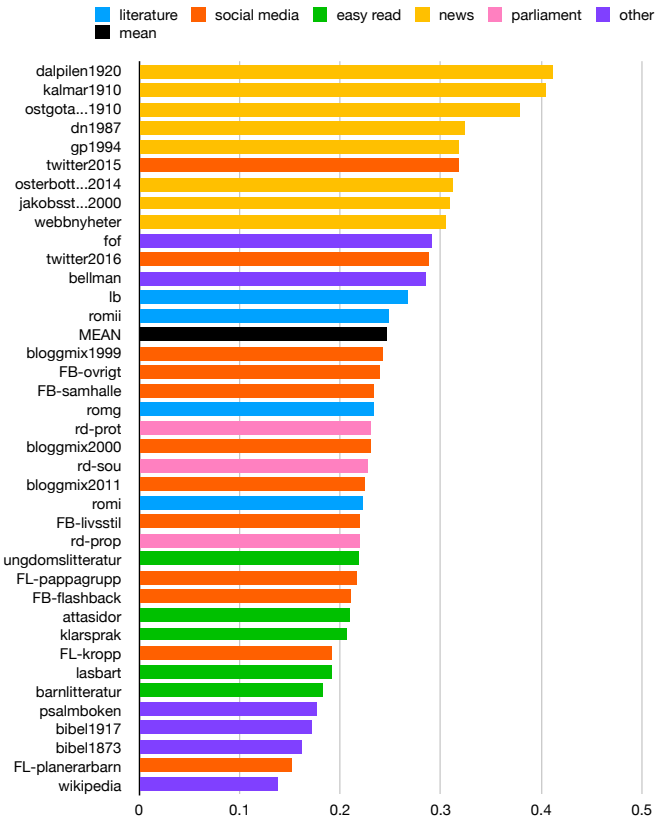


Figure 12: TTR at $n = 10\,000$

Figure 12 shows the results for $n = 10\,000$, on 38 corpora. We see that it fairly well separates several categories of text. The eight newspaper corpora are above all but one other, with the three oldest getting the highest value, followed by the two from the late 1900s, then the two from printed newspapers in 2000 and 2014, and last the web-based news texts. (The difference may be partially explained by OCR errors.) The social media and blog texts are a little more scattered, but all below the mean, except Twitter, which in both cases is higher. The four corpora of novels are not quite the same level, but all higher than all of the ones in the "easy read" category. In that category, young adult literature is the highest and children's literature the lowest. Parliamentary data is all below the mean but above "easy read". Near the bottom we find, perhaps surprisingly, the Bible, along with Wikipedia, neither of which are primarily known to be easy reads. Altogether, these results should tell us that this is at least a meaningful measure.

That leaves the question of choosing an n. Very low values might give strange effects, very high values would make it unusable for shorter texts. Other values were tested for comparison: $n = 10$ gives little useful information, while $n = 100$ ranks all the novels below most of social media, and beyond that we get mostly unremarkable results from just looking at the ranking. Based on these limited results, $n = 10\,000$ is a good choice, and for short texts we can settle for $n = 1000$.

6 Spectrum comparison

Instead of considering type counts for only one n, what if we measure for many values of n, and look at the whole spectrum? This is essentially what we already did in all of section 3, and we could see that the curves for the different corpora certainly did have different shapes – some of them even crossed each other, which implies that any one number is not going to tell us the whole truth.

To compare corpora instead of methods, we need to pick one method, one way to transform u based on n. Using plain TTR as seen in Figure 1 would make it difficult to tell the difference between shapes, and picking one of the tested methods seems like too arbitrary a choice. So for the purposes of this section, we will evade the problem. We normalise the type count (or equivalently TTR) for each n by subtracting the mean and dividing by the standard deviation. That is, the values on the vertical axis are in terms of standard deviations above the mean, counted for each separate value on the horizontal axis. (For high values, the mean/sd change erratically because of corpora dropping off. We adjust the normalisation to gradually change from actual to extrapolated mean/sd.)

Figures 13-22 show the spectra for each category. Some curves are shorter because of limited data. Figures 13-15 show different types of web-based texts, one set of blog texts and two different internet forums. We can see that each category is a little different, but all the curves share some characteristics – a short rise, then a drop, then flatter, and finally a small rise. Most of them start slightly above the mean, and end below the mean.

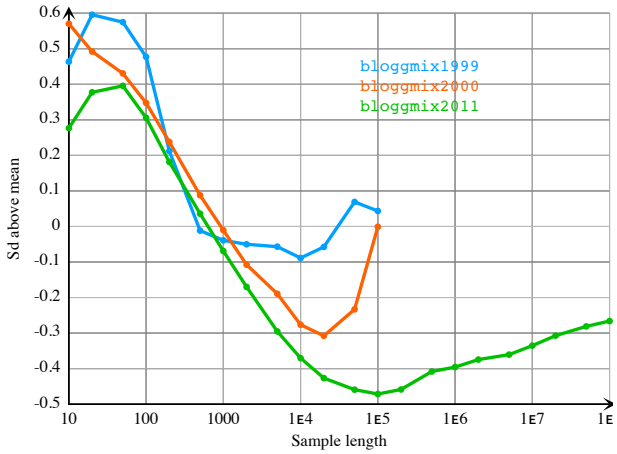


Figure 13: Spectrum for blog texts

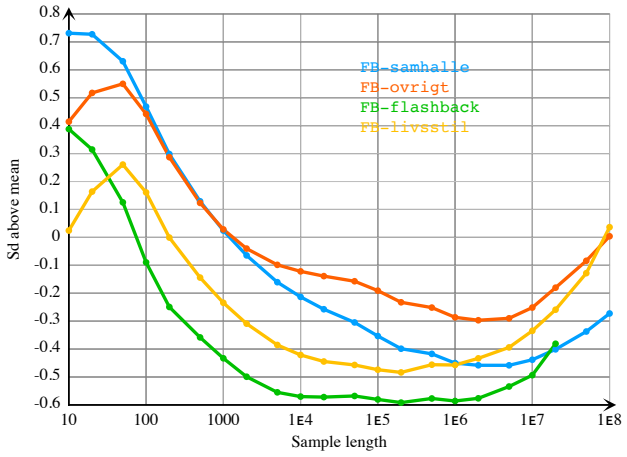


Figure 14: Spectrum for the *Familjeliv* forum

Figure 16 shows the "easy read" category. Despite being unrelated, the curves share the same shape, which is clearly different from the web-based corpora – a drop, then a rise, peaking around 1000 without reaching the mean, then a drop.

Figures 17-18 show news texts, with Figure 17 showing three newspapers from the early 1900s,

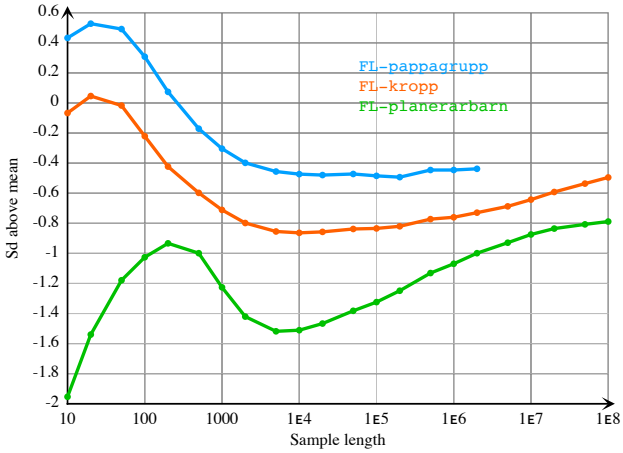


Figure 15: Spectrum for the *Flashback* forum

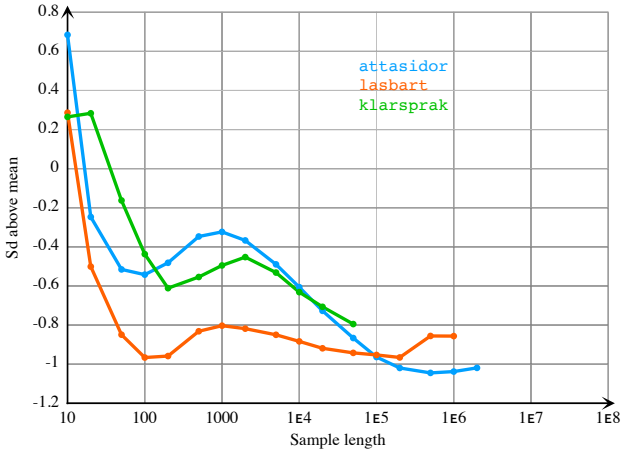


Figure 16: Spectrum for easy-read texts

and Figure 18 showing four more recent newspapers and one web-based news corpus. As with the blog/forum collection, we see that these two related categories have clear similarities: a slow rise up to between 10 000 and 100 000, and then a drop. But they are also visibly distinct, with the older newspapers having higher values and rising near the end. Aside from some unpredictable behaviour for $n < 1000$, the curves in each category are remarkably similar in both shape and level.

Figures 19-20 show literary texts, with Figure 19 showing regular novels and Figure 20 showing children's fiction and young adult fiction. They are all comparatively straight and dropping slightly. Children's literature is generally lower than young adult literature, and they both drop faster than the curves for books aimed at adults.

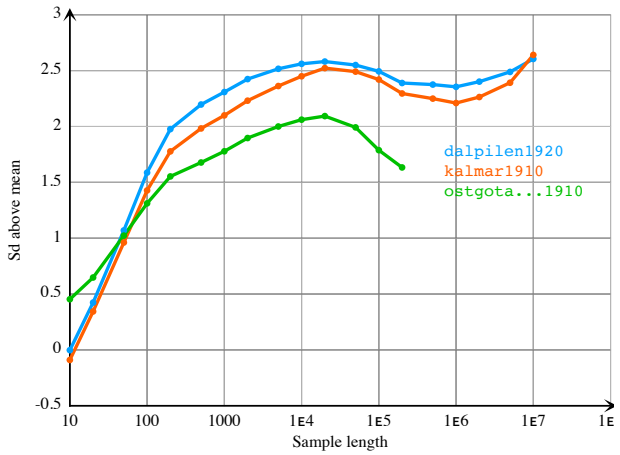


Figure 17: Spectrum for old newspapers

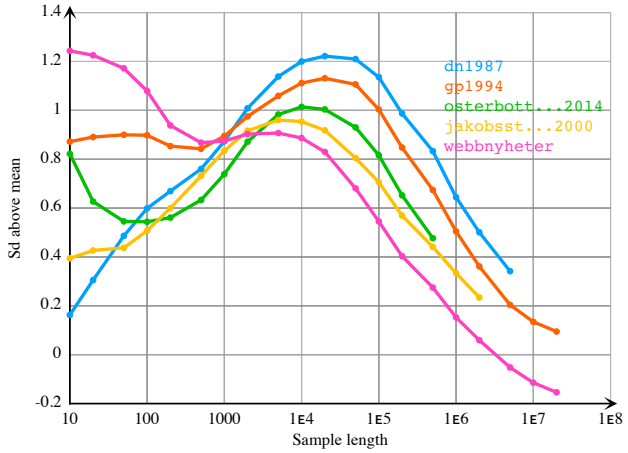


Figure 18: Spectrum for recent newspapers

Figure 21 shows religious texts. We see two translations of the Bible, with very similar curves – both dropping, rising, levelling out, but unlike the easy read category they level out at about the same level where they started. Also included is a book of church hymns, which happens to level out at a similar level, but starts with a large rise.

Finally, in Figure 22, we see three uncategorised corpora – one from a 1700s songwriter, one from a popular science magazine, and one from Wikipedia. As expected, they show very different shapes and levels, and are clearly distinct from each other as well as all the other curves.

Explanations of the shapes are tentative at this point, but we can guess at the meaning of high richness in different regions of sample length: For low values (roughly 100-1000), it may indicate complex sentences with few function words; for medium values (around 10 000), complexity in topics, with many names etc.; and for high values, variety in topics. This may explain why newspapers peak in the middle (they address complex topics with many names, but return to the same topics), social media drop in the middle (they address simpler topics but with a wider variety in topic and style), and youth novels go down (they are on simple topics and consistent for entire books). Further speculation is left for future work.

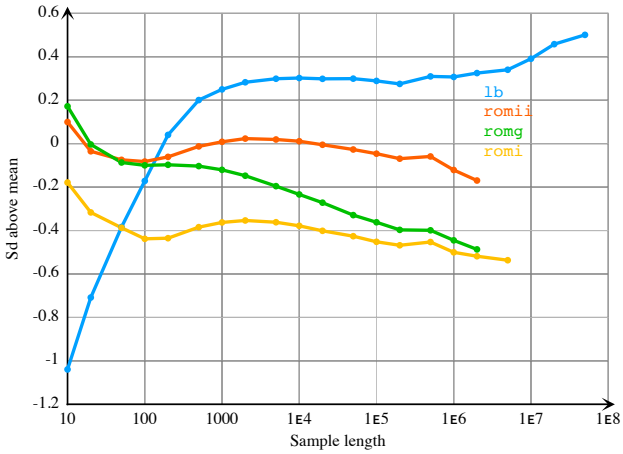


Figure 19: Spectrum for novels

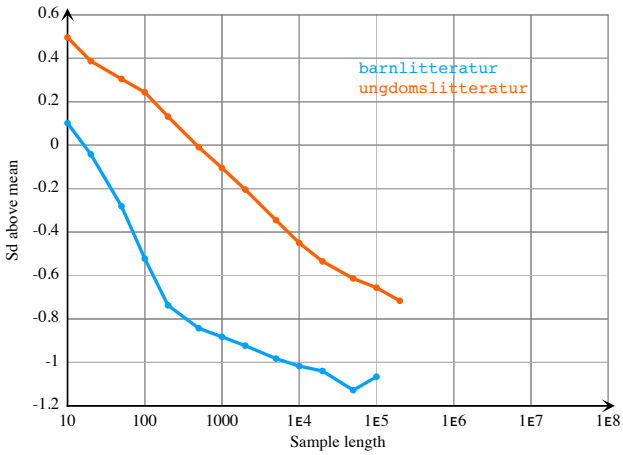


Figure 20: Spectrum for youth novels

7 Applicability

Is it reasonable to apply measures like these on an entire corpus instead of just separate texts? First, "separate texts" is not necessarily well defined. Is

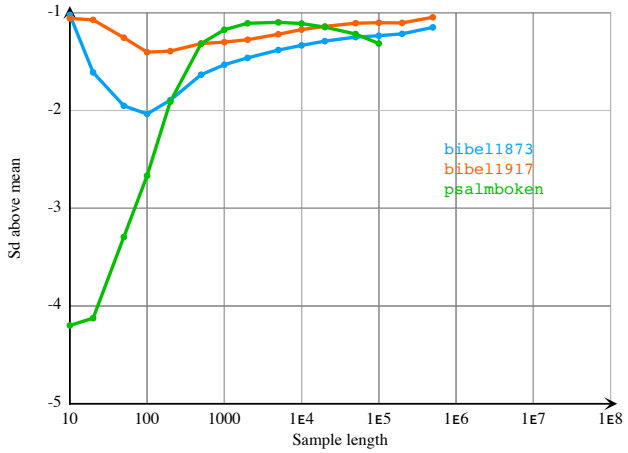


Figure 21: Spectrum for religious texts

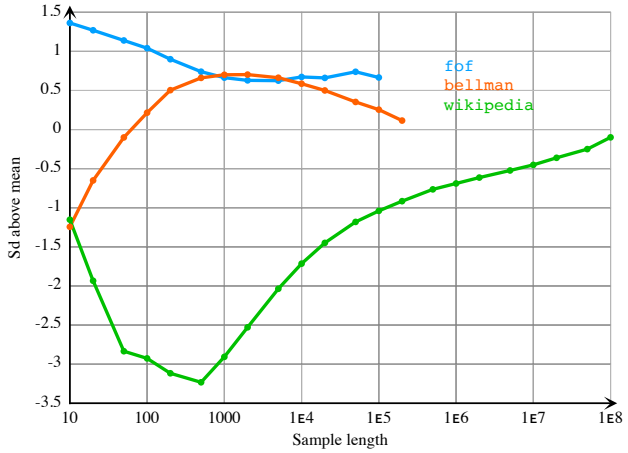


Figure 22: Spectrum for some other texts

a newspaper one text, or each article? Books in a series? Multiple entries on a web page? Second, for low values of n, running the entire corpus at once should make little difference. For example, if $n = 100$ and the typical length of a text is 10 000, only about 1% of samples would contain two texts, and the rest only one. For high values of n, using only separate texts would leave us with no data at all – it would be difficult to find singular coherent texts spanning hundreds of millions of words. This means that allowing corpora of multiple authors and topics is our only option for large n.

But we can also look at the results. Are the differences between the curves largely caused by differences in text length? If that was the case, we would expect that when a curve reaches the "critical n" where we go from a single text to multiple texts, the vocabulary richness should increase rapidly. The curve we would expect to see is one that starts out mostly flat (because hardly any texts are that short), then slowly decreases (as others reach their critical n and bring up the mean), then rapidly jumps up as it reaches its critical n, and then slowly decreases again. This is not a pattern that we see anywhere, so we can conclude that text length is not the driving factor of the curve shapes.

8 Conclusion

The task of finding a length-independent measure of vocabulary richness is difficult at best. We have seen that many traditional measures are not satisfactory, and made some suggestions as to how they can be improved. Perhaps the most obvious approach is to use average TTR over a sample length, with 10 000 words being a good sample length.

The figures show that the curves have very different shapes, and often cross. Thus, the ranking of corpora changes depending on the length of the text sample, so a perfect solution is not possible, or at least cannot be expressed as a single number.

Is this spectrum method useful for genre classification? It is perhaps rare that we need to analyse entire hundred-million-word corpora to see if they are made up of novels or newspapers, but we do see some differences even for much smaller lengths. We have also gained insight into what makes it difficult to find a good measure of vocabulary richness. Most importantly, we have seen that there are notable differences between genres, and raised for future research the question of why.

References

Etienne Brunet. 1978. *Le vocabulaire de Jean Giraudoux structure et évolution*. Slatkine, Genève.

Daniel Dugast. 1979. *Vocabulaire et stylistique*, volume 8. Slatkine, Genève.

Gustav Herdan. 1964. *Quantitative linguistics*. Butterworth, London.

Tor G. Hultman and Margareta Westman. 1977. *Gymnasistsvenska*. Liber Läromedel, Lund.

Heinz-Dieter Maas. 1972. Über den zusammenhang zwischen wortschatzumfang und länge eines textes. *Zeitschrift für Literaturwissenschaft und Linguistik*, 2(8):73.

Fiona J Tweedie and R Harald Baayen. 1998. How variable may a constant be? Measures of lexical richness in perspective. *Computers and the Humanities*, 32:323–352.

Appendix A. List of corpora

The following corpora were used, all of which can be found at spraakbanken.gu.se/en/resources (some only in scrambled versions). All texts are in Swedish.

attasidor issues of the newspaper *8 sidor* in easy Swedish
barnlitteratur collection of children's literature
bellman lyrics from the Swedish songwriter C. M. Bellman (1740-1795)
bibel1873 full text of the 1873 Swedish Bible translation
bibel1917 full text of the 1917 Swedish Bible translation
bloggmix1999 collection of blog texts from 1999
bloggmix2000 collection of blog texts from 2000
bloggmix2011 collection of blog texts from 2011
dalpilen1920 issues of the newspaper *Dalpilen* from the 1920s
dn1987 issues of the newspaper *Dagens Nyheter* from 1987
familjeliv (FL) kropp webforum *familjeliv.se*, subforum about the human body
familjeliv (FL) planerarbarn webforum *familjeliv.se*, subforum about planning to have children
familjeliv (FL) pappagrupp webforum *familjeliv.se*, subforum for fathers
flashback (FB) flashback webforum *flashback.se*, subforum about the forum itself
flashback (FB) livsstil webforum *flashback.se*, subforum about lifestyle
flashback (FB) ovrigt webforum *flashback.se*, subforum about miscellaneous topics
flashback (FB) samhalle webforum *flashback.se*, subforum about society
fof issues of the popular science magazine *Forskning & Framsteg*
gp1994 issues of the newspaper *Göteborgsposten* from 1994
jakobstadstidning2000 issues of the newspaper *Jakobstads Tidning* from 2000
kalmar1910 issues of the newspaper *Kalmar* from the 1910s
klarsprak administrative authority texts
lasbart collection of easy-read texts and children's books
lb the Swedish Literature Bank, a collection of literature mainly from around 1900
osterbottenstidning2014 issues of the newspaper *Österbottens Tidning* from 2014
ostgotaposten1910 issues of the newspaper *Östgötaposten* from the 1910s
psalmboken the hymn book of the Church of Sweden
rd-prop Swedish parliament texts, propositions
rd-prot Swedish parliament texts, protocols
rd-sou Swedish Government Official Reports
romg collection of older novels
romi collection of modern novels
romii collection of modern novels
twitter-2015 posts from twitter.com, 2015
twitter-2016 posts from twitter.com, 2016
ungdomslitteratur young adult literature
webbnyheter2005 collection of online newspaper texts from 2005
wikipedia Swedish Wikipedia, collected in 2017

A query engine for L1-L2 parallel dependency treebanks

Arianna Masciolini
Språkbanken Text
Department of Swedish, Multilingualism, Language Technology
University of Gothenburg
arianna.masciolini@gu.se

Abstract

L1-L2 parallel dependency treebanks are learner corpora with interoperability as their main design goal. They consist of sentences produced by learners of a second language (L2) paired with native-like (L1) correction hypotheses. Rather than explicitly labelled for errors, these are annotated following the Universal Dependencies standard. This implies relying on tree queries for error retrieval. Work in this direction is, however, limited. We present a query engine for L1-L2 treebanks and evaluate it on two corpora, one manually validated and one automatically parsed.

1 Introduction

L1-L2 parallel dependency treebanks are learner corpora where sentences produced by learners of a second language (L2) are paired with correction hypotheses, assumed to be native-like and therefore referred to as *L1 sentences*. Both the learner originals and the corresponding corrections are annotated following the cross-lingual Universal Dependencies (UD) standard (Nivre et al., 2020). The idea is that such morphosyntactical information makes explicit error labelling unnecessary and allows errors to instead be retrieved via tree queries. This format, proposed by Lee et al. (2017a), was in fact designed to address the interoperability issues arising from the coexistence of the different markup styles and error taxonomies normally employed for the annotation of learner corpora. These tend not only to be language-specific, but also to vary widely across different same-language projects. An additional advantage of using UD is the availability of several increasingly fast and reliable parsers (Straka, 2018; Qi et al., 2020). While not yet very robust to learner errors (Huang et al., 2018), they can already speed up the annotation process significantly.

L1-L2 UD treebanks exist for English (Berzak et al., 2016), Chinese (Lee et al., 2017b) and Italian (Di Nuovo et al., 2022). Work on error retrieval tools, on the other hand, has been limited. Only one of these corpora, the ESL (English as a Second Language) treebank, is equipped with a query engine.[1] This tool, however, presents several limitations, a major one being its reliance on a pre-existing error taxonomy, in contrast with Lee et al. (2017a)'s idea.[2] Closer in spirit to the latter, Choshen et al. (2020) have developed a method to automatically derive dynamic syntactical error taxonomies from L1-L2 treebanks, but do not provide a way to look for specific error patterns.

In this paper, we present a language- and error taxonomy-agnostic query engine for L1-L2 parallel dependency treebanks. The tool allows searching for morphosyntactical errors by describing them in a pre-existing pattern matching language for UD trees, which we extend to facilitate comparing L2 sentences to their corrections, resulting in what we call *L1-L2 patterns*. Our main contribution is a sentence retrieval algorithm that matches the L1 and L2 portions of a query pattern on the corresponding treebanks in parallel, ensuring that correspondences are found between segments that align with each other. Addressing another limitation of the existing tools, we also make it possible to extract the specific portions of an L1-L2 sentence that match a given pattern. The engine is part of L2-UD, a larger open source toolkit for UD-annotated L2 data, available for download at github.com/harisont/L2-UD.[3]

[1] As of 05.04.2023, the ESL treebank's homepage, esltreebank.org, seems to be no longer reachable, but the user interface of the query engine can be inspected at the Internet Archive: web.archive.org/web/20220120204838/http://esltreebank.org.

[2] Note, however, that the ESL treebank actually predates Lee et al.'s paper.

[3] The results reported in this paper were obtained with version 0 of the engine: github.com/harisont/L2-UD/releases/tag/v0 (last access 05.04.2023).

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 574–587

Figure 1: UD trees for the Italian sentence *Sono andat{o→a*} da entrambi* ("I have been to both"), discrepancies highlighted in bold. In the L2 sentence, displayed on the right, the gender of the participle *andato*, referring to the implicit subject of the sentence, is incorrect, but without further context we have no way to infer the author's gender. As a consequence, the error can only be described in terms of its correction. Example adapted from the VALICO-UD treebank.

2 Related work

As mentioned in the introduction, learner corpora exist in a variety of formats. Lee et al. (2017a)'s proposal to use L1-L2 parallel dependency treebanks is not the only one aimed at overcoming the interoperability issues that follow. Bryant et al. (2017), for instance, introduced ERRANT, an ERror ANnotation Toolkit operating in the framework of a taxonomy that exclusively relies on dataset-agnostic information such as the POS (Part Of Speech) tag and morphological features of the tokens involved. In a sense, this can be seen as an attempt to solve the problem by developing a "universal" error taxonomy. While ERRANT has become dominant in Grammatical Error Correction research, it still coexists with several other tagsets which differ significantly both in their underlying assumptions, often language-specific, and in the granularity of the annotation, which varies according to the intended use of each individual corpus.

Lee et al.'s idea, while only concerned with morphosyntactical errors, is more radical, as UD-annotated parallel treebanks have the potential to remove the need for any explicit error pre-categorization and instead allow to infer error taxonomies automatically and dynamically. Choshen et al.'s work on syntactical error classification, showing promising results even on automatically parsed L1-L2 treebanks, goes in this direction.

While there is a wide variety of tools and language to choose from for extracting information from (monolingual) UD treebanks,[4] not many options are available when it comes to retrieving example sentences matching specific patterns of error from L1-L2 treebanks. To the best of our knowledge, the above mentioned ESL treebank query engine is the only tool specifically meant for this task. While it is reasonable to assume that the latter could easily be generalized to work with any L1-L2 treebank, it presents several limitations from the perspective of the tree queries envisioned by Lee et al. (2017a). First and foremost, searching for errors is primarily done by selecting an error label from a pre-defined set. A simple query language is also available, but it only allows searching for sequences of word forms, POS tags and dependency labels. In other words, UD sentences are treated as lists of tokens rather than trees. This can be restrictive since, for the purposes of grammatical error retrieval, dependency structure is often more relevant than linear order. Furthermore, there is no coupling, other than sentence-level alignment, between the L1 and L2 parts of the treebank. Patterns are therefore only matched against L2 sentences, making it impossible to search for errors whose description requires a comparison with the correction (cf. Figure 1 for an example) or locate the relevant portions of their L1 counterparts. A final, related limitation is that the tool always returns complete sentences, while it is sometimes useful to isolate the segments that match the query.

3 Design and implementation

Addressing these limitations, we aim for a query engine with the following characteristics:

1. **no underlying error taxonomy**: errors are described in a pattern matching language which allows treating UD sentences both as sequences of tokens and as tree structures;
2. **parallel L1-L2 matching**: queries consist in an L1 pattern, that has to be matched by a correction hypothesis, and an L2 pattern to be matched in the corresponding learner sentence. This also allows formulating queries by comparing learner sentences with their corrections;
3. **subsentence extraction**: besides retrieving full sentence pairs, it is also possible to extract the specific portions of an L1-L2 pair actually matching the query.

[4] PML-TQ (Pajas and Štěpánek, 2009), GREW-MATCH (Guillaume, 2021), SETS (Luotolahti et al., 2015) and TÜNDRA (Martens, 2013), just to name a few.

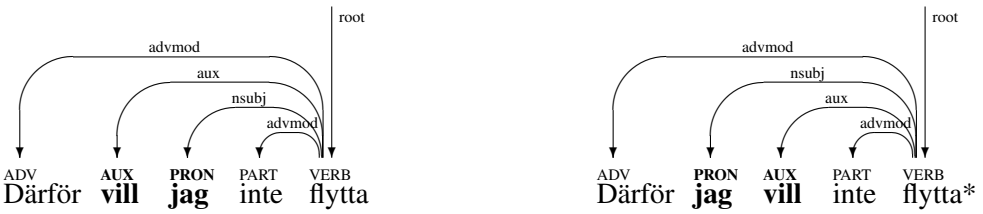


Figure 2: UD trees for the L1-L2 Swedish sentence *Därför {vill jag → jag vill*} inte flytta* ("Therefore I don't want to move"), discrepancies highlighted in bold. The L2 sentence, on the right, violates V2 word order. Example adapted from the DaLAJ corpus.

3.1 Query language

A central part of the engine is its query language. We start with an overview of the pre-existing pattern matching language our system makes use of. After that, we present the extensions through which we adapt it to querying L1-L2 treebanks.

3.1.1 UD patterns

To describe morphosyntactical structures, we use the Haskell-embedded pattern matching language available as part of the GF-UD toolset for dependency trees and interlingual syntax (Kolachina and Ranta, 2016; Ranta and Kolachina, 2017).[5] Although not as widespread as some of the above mentioned alternatives, it allows to express a wide range of queries with an intuitive syntax, and it was selected due to its ease of integration with the other components of the project.

In essence, the language provides three types of patterns:

- *single-token patterns*, e.g. `POS "VERB"`, matching all (sub)trees rooted in a verb. With a similar syntax, it is possible to pattern match based on the token's `XPOS`, `DEPREL`, `FEATS`, `FORM` or `LEMMA`, all of which correspond to homonymous CoNNL-U fields;[6]
- *tree patterns*, in the form `TREE p [ps]`, where `p` is a pattern to be matched by the root node and `[ps]` an ordered list of patterns denoting its dependents. For instance, the pattern `TREE (POS "VERB") [DEPREL "nsubj", DEPREL "obj"]` matches all subtrees rooted in a verb having exactly two subtrees: a nominal subject `nsubj` and a direct object `obj`, in this order;
- *sequence patterns*, matching subtrees where a certain sequence of patterns occurs with no intervening words. For instance, in Subject-Verb-Object (SVO) languages we might want to write `SEQUENCE [POS "VERB", DEPREL "nsubj", DEPREL "obj"]`.

More liberal versions of some of these patterns, using the original name followed by an underscore, also exists. Namely, `DEPREL_ d` ignores relation subtypes, `FEATS_ fs` matches all tokens whose morphological features include (rather than conicide with) `fs`, `TREE_ p [ps]` allows other dependents to appear before, between and/or after the explicitly listed ones and `SEQUENCE_ ps` does not require the listed patterns to occur contiguously. In addition, the language allows to combine patterns with the logical operators `AND`, `OR` and `NOT` and provides a `TRUE` pattern matching any subtree.

As a slightly more complex example, consider

```
TREE_
  (POS "VERB")
  [DEPREL_ "nsubj",
   OR [DEPREL "obj", DEPREL "obl"]]
```

The above pattern matches any subtree rooted in a verb which has at least two dependents: a nominal subject (ignoring any subtyping) and a direct object `obj` or oblique `obl` (not subtyped).

3.1.2 L1-L2 patterns

In some cases, errors can be described by a single UD pattern to be looked for in the L2 treebank. Often, however, it is more convenient and concise (if not even necessary, as illustrated in Figure 1) to describe errors by comparing an L2 sentence to its correction. For this reason, queries in our system are defined as pairs of UD patterns. This, however, does not prevent writing queries as L2-only patterns: any single-pattern query `q` is simply expanded to a pair $\langle$`TRUE`, `q`$\rangle$.

As an example of the usefulness of L1-L2 patterns, consider the sentence displayed in Figure 2: in the L2 text displayed on the right, the learner is

[5] For an exhaustive description of the pattern matching language, see the relevant GF-UD documentation: `github.com/GrammaticalFramework/gf-ud/blob/master/doc/patterns.md` (last access 05.04.2023).

[6] For an overview of the CoNNL-U format and a complete list of the abbreviations used in this text, see Appendix A.

using Swedish's default SVO order, with the pronoun *jag* preceding the auxiliary verb *vill*. The sentence, however, starts with the adverb *därför*. Being Swedish a language with verb-second (V2) word order, the correction, displayed on the left, moves the auxiliary in the second position, right after the adverb itself. A way to find L2 sentences presenting the same problem is to use a single sequence pattern, for instance:

```
SEQUENCE [
  POS "ADV",
  OR [POS "VERB", POS "AUX"],
  DEPREL_ "nsubj"]
```

This does match sentences like the one above, but does not cover all cases in which V2 order is violated: rather than with an adverb, the sentence might for example start with a prepositional phrase (cf. *På grund av detta vill jag inte flytta*, with the similar meaning of "Because of this I don't want to move"). Rather than enumerating all possible patterns of V2 order violation, it can be more convenient to express the error in terms of its correction, for instance with the following pair of patterns, which disregards the opening phrase:

```
L1 : SEQUENCE [
       OR [POS "VERB", POS "AUX"],
       DEPREL "nsubj"]
L2 : SEQUENCE [
       DEPREL "nsubj",
       OR [POS "VERB", POS "AUX"]]
```

For the sake of conciseness, rather than writing two separate patterns, we enclose the discrepant portion in curly brackets and divide the L1 and L2 segments with an arrow:

```
SEQUENCE [
  {OR [POS "VERB", POS "AUX"],
  DEPREL "nsubj" →
  DEPREL "nsubj",
  OR [POS "VERB", POS "AUX"]}]
```

This is our first extension to the pattern matching language described in Section 3.1.1.

To avoid repetition, we also introduce variables. As an example use case, consider gender agreement, a source af confusion for learners of many languages. In a dependency tree, most errors of this kind can be identified by checking whether the gender of certain dependents matches the gender of the token they are referred to. In Italian, for instance, adjectives should agree with the nouns they modify. This is not the case in the sentences like *Indossava una maglietta nero* ("(S)he was wearing a black t-shirt"), where the noun, *maglietta*, is feminine, while the adjective is incorrectly inflected in its masculine form *nero*. This particular sentence therefore matches the pattern

```
TREE_
  (FEATS_ "Gender=Fem")
  [AND [DEPREL "amod",
        FEATS_ "Gender=Masc"]]
```

With the syntax presented until now, however, looking for all noun-adjective gender agreement errors requires a separate query for each possible combination of genders.[7] With variables, syntactically characterized by capital letters preceded by a `$` sign, we can instead simply write

```
TREE_
  (FEATS_ "Gender=$A")
  [AND [DEPREL "amod",
        FEATS_ "Gender=$B"]]
```

where `$A` is assumed to be different from `$B`. Variables are currently supported for morphological features, Universal POS tags and dependency relations, all of which have a finite number of possible values.

3.2 Sentence retrieval algorithm

Alongside the pattern matching language, GF-UD provides a function that, given a pattern and a UD tree, recursively checks if the former matches the latter itself or any of its subtrees. One might be prone to think that performing an L1-L2 query can simply consist in applying this function to all trees in the treebank, looking for L1 sentences matching the L1 portion of the pattern and L2 sentences matching its L2 portion. Doing that, however, generally leads to a significant amount of false positives. Consider, for instance, the following query, intended for searching number agreement errors between a head and its direct dependents:

```
TREE_
  (FEATS_ "Number=$A")
  [FEATS_ "Number={$A → $B}"]
```

Following this naïve approach, the sentence in Figure 1 would match the pattern even if it does not contain a number agreement error. This happens because the L1 sentence matches the L1 pattern at *sono andato* ("(I) have been", two singular verb forms), while the L2 sentence matches the L2 pattern at *andata da entrambi* ("been to both"), where the head *andata* is again a singular but the dependent *entrambi* is a pronoun in its masculine plural form. In this case, in fact, both the original

[7] Two in Italian, whose only genders are masculine and feminine, but already six for languages with neuter!

sentence and its correction match both portions of the pattern.

A key observation here is that *sono andato* and *andata da entrambi* do not semantically correspond to each other: to solve the problem, we need to further align our L1-L2 treebank, recursively putting L1 subtrees in correspondence with their L2 counterparts. To do that, we use the CONCEPT-ALIGNMENT Haskell library (Masciolini and Ranta, 2021). While originally designed for extracting translation equivalents from multilingual parallel treebanks, its *alignment criteria*, i.e. the set of rules to decide whether two subtrees correspond to each other, are configurable and easy to adapt to the L1-L2 domain. Actually, accuracy on L1-L2 corpora tends to be better than it is for multilingual treebanks. Learner sentences and their corrections, in fact, usually share the vast majority of the lemmas, something that can be taken into account when defining custom alignment criteria.

As a first step, then, we extract phrase- and word alignments, in the form of pairs of L1-L2 UD trees, for each L1-L2 sentence pair. After that, to decide whether a sentence pair matches a given L1-L2 pattern, we apply a nonrecursive version of GF-UD's pattern matching function to check if there is a pair of aligned subtrees whose L1 (resp. L2) component matches the L1 (resp. L2) portion of the pattern. Matching nonrecursively, only on complete UD trees (altough extracted from full sentences), is crucial here, as it is, in most cases, what prevents L1-L2 patterns from being matched in structurally similar but semantically unrelated subtrees of the L1-L2 sentence pair.

The careful reader, however, will have noticed that this does not solve the issue for the specific example mentioned, where the subtrees matching the L1-L2 pattern, *sono andata* and *andata da entrambi*, share the same head *andata*. The false positive is due to the fact that its dependents, *sono* and *da entrambi*, do not correspond to each other. For `TREE` and `TREE_` patterns, then, we recursively perform the additional check that all dependents involved in the match are aligned with each other. A similar mechanism is in place for `SEQUENCE` and `SEQUENCE_` patterns, to avoid matching subsequences that, while part of the same subtree, are not semantically equivalent.

3.3 Subsentence extraction

By default, the output of the program is the list of IDs of the sentences matching the given query. Nontheless, extracting relevant subsentences can be useful both for futher processing of the error-correction pair and to more easily visualize discrepancies in the context in which they occur.

The fact that our sentence retreival algorithm applies patterns on sub-sentence alignments makes it straightforward to locate the specific L1 and L2 subtrees where the match is found. Doing so, however, is of very limited usefulness when the root (or the head of a large subtree) is involved in the error, resulting in too big subtree pairs. For this reason, we prune the extracted subtrees by only keeping the portions explicitly described by the pattern: individual heads for single-token queries, heads and their dependents that match a pattern in `ts` for `TREE_ ts` patterns and, for sequence patterns, rather than the whole subtree including the given sequence, only subtrees matching one of patterns explicitly listed in it. Implementation-wise, this is done by converting the query's UD patterns into *replacement patterns* in GF-UD's tree manipulation language.[8]

The engine has options to either extract such pairs of matching subsentences and write them to CoNNL-U files or to output a Markdown report where they are highlighted in the sentences where they occur. Example reports obtained with the latter method can be found in Appendix C.

4 Evaluation

Aiming at assessing the performance of the query engine, we tested it on two L1-L2 error-tagged corpora in two different languages, one that comes with manually validated UD annotation and one that was only parsed automatically. In both cases, we randomly selected 100 sentences to be used during development and set the rest aside for testing. Carrying out a systematic evaluation was not possible: more often than not, an error tag maps not to a single L1-L2 query, but to a potentially rather large set of queries whose exhaustiveness is hard to verify. As a consequence, we opted for computing the sentence-level precision and recall obtained upon running, for each corpus, a

[8]The pattern replacement language is, in many ways, analogous to the pattern replacement language and documented alongside it: `github.com/GrammaticalFramework/gf-ud/blob/master/doc/patterns.md` (last access: 05.04.2023).

single-token, a tree and a sequence example query, all chosen to be descriptive of an error typical of the language at hand. To automate evaluation as much as possible, we also tried to make each query match one of the error labels of the dataset at hand. In this way, performance for a given query can be assessed by simply comparing the sentence IDs returned by the engine with those of the sentences marked with the corresponding error label. Finding exact correspondences was feasible for single-token queries, but challenging for tree and sequence patterns, which tend to be finer-grained. In such cases, we formulated queries describing a subset of the error cases denoted by a certain label and manually inspected sentences marked with it to select the relevant items. By comparing the results obtained on the two corpora, we also aim at getting insights about the ways in which automatic annotation affects the performance of the engine, even though we cannot quantify its impact.

4.1 Experiments on manually validated data

4.1.1 Data

Our first treebank of choice is the 398-sentence manually validated subset of the VALICO-UD corpus (Di Nuovo et al., 2022), consisting of texts written by Italian L2 learners with various L1 backgrounds. While much smaller than the above mentioned ESL treebank, it was deemed preferable due to its more complete UD annotation.[9]

Error tagging, present as sentence metadata, is XML-like and based on Nicholls (2003), where each label consists of a two-letter code, with the first character representing the general class of error (inflection, omission etc.), and the second generally specifying the POS tag of the word(s) involved. In some cases, VALICO-UD labels also present a third letter, usually denoting an incorrect inflectional feature.[10] In the error tag IDG, for instance, the three letters stand for "Inflection", "Determiner" and "Gender" and are meant to enclose determiners incorrectly inflected for gender.

[9]Due to licensing issues, the UD annotation of the ESL corpus is released separately from the learner essays themselves. Consequently, in order to prevent the text from being reconstructed from the annotation, the LEMMA and FEATS fields are left blank.

[10]An exhaustive description of the error annotation guidelines is given at raw.githubusercontent.com/ElisaDiNuovo/VALICO-UD_guidelines/main/Error_Annotation_Guidelines_v.1.1.pdf (last access: 05.04.2023).

	Precision	**Recall**
V_1	43% (40%)	100%
V_1'	100% (90%)	100% (64%)
V_2	100%	40%
V_3	-	0%
V_3'	100%	100%

Table 1: Precision and recall of the example queries run on the VALICO-UD corpus. Values in parentheses do not take error annotation issues into account.

4.1.2 Queries

Gender being a notorious source of confusion for learners of Italian, we chose the set of two L1-L2 patterns equivalent to IDG as our first test query:

```
V1: AND [
      POS "DET",
      FEATS_ "Gender={$A → $B}"]
```

A second, more complex query is

```
V2: TREE
      (POS "NOUN")
      [ {DEPREL "det", → }
        DEPREL "det:poss" ]
```

which denotes a subclass of the MD (Missing Determiner) VALICO category describing the common error pattern for which the definite article that should precede a possessive modifying a noun is omitted (consider, for instance, the nominal phrase *{il → _*} suo naso* - "his nose").

Word order is relatively free in Italian, and finding recurrent patterns in such a small corpus is not easy. Instead, we use a sequence pattern to find particular occurrences of RD (Replacement of Determiner) errors:[11]

```
V3: SEQUENCE [
      LEMMA "non",
      LEMMA "ci",
      LEMMA "essere",
      LEMMA {"nessun*" −> "un*"}]
```

This pattern matches phrases that translate to "there is/are no x". In Italian, this is usually expressed with a double negation: not only is there an initial negation, *non*, but the determiner introducing x (*nessuno* or *nessuna*, depending on x's gender) also has negative polarity. However, since this is not the case in most other languages, it is common for L2 speakers to simply use the indefinite article (*un/un'/uno/una*).

4.1.3 Results

As displayed in Table 1, recall is perfect for the first query. The low precision should not mis-

[11]Even though UD patterns do not support general regular expressions, an asterisk at the beginning or end of a string can be used as a wildcard.

lead: one of the false positives is due to an inconsistency in the error annotation and 8 of the remaining 20 false positives are due to cascading errors. It is often the case, in fact, that the gender of the noun the determiner is referred to is wrong, and the incorrect inflection of the determiner introducing it is merely a consequence of that. In this case, the incorrect noun is marked with the RN (Replace Noun) label in case of a lexical error (cf. *la panca* → *il banco**, "the bench → the desk") or with the ING (Incorrect Noun Gender) label if the noun is incorrectly inflected (cf. *gli uccelli* → *le uccelle**", "the birds"), while the determiner gets the IDGcascade tag. To avoid matching cascading errors, we can turn V_1 into a TREE query and check whether the determiner's gender agrees with the noun's:

```
V1': TREE_
     (AND [POS "NOUN",
           FEATS_ "Gender=$A"])
     [AND [POS "DET",
           FEATS_ "Gender={$A → $B}"]]
```

As Table 1 shows, the precision for V_1' is significantly higher. While recall appears to decline, all of the false negatives can be traced back to errors, inconsistencies or incompletenesses in either UD annotation or error tagging (see Appendix B for a complete list of issues found in the VALICO-UD corpus).

When it comes to V_2, there are no false positives, while the 3 false negatives are due to alignment errors. In every case, the sentence at hand presents several errors, so that the L1 and L2 trees differ significantly, increasing the difficulty of the alignment task.

V_3, on the other hand, only has one expected hit, the sentence *-Non c'è {**nessun** bacio → **una** baciata*} per me,- ha pensato tristemente.* (-There's no kiss for me,- (s)he thought sadly."), but no matches. Again, the problem seems to be an alignment error, since the expected sentence id is indeed returned if we use a similar L2-only query:

```
V3': SEQUENCE [
       LEMMA "non",
       LEMMA "ci",
       LEMMA "essere",
       LEMMA "un*"]
```

4.2 Experiments on parsed data

4.2.1 Data

To evaluate the tool on automatically annotated data, we used a 2087-sentence subset of the DaLAJ corpus (Volodina et al., 2021).[12] Such corpus is composed of L1-L2 sentence pairs automatically derived from the error-annotated learner corpus of anonymized L2 Swedish essays SweLL (Volodina et al., 2019). More specifically, SweLL essays are processed so that the L2 component of each sentence pair in the DaLAJ corpus contains exactly one morphological or syntactical error. Arguably, this makes automatically parsing the L2 sentences and aligning them to their L1 counterparts significantly easier than it would be if multiple, possibly cascading and/or overlapping errors coexisted. Evaluating the tool on the SweLL corpus itself, however, would have been extremely impractical, as the original versions of the essays are not sentence-aligned.

In terms of error-annotation, since DaLAJ entries only contain one error each, sentence pairs are simply assigned a SweLL error label. SweLL's error taxonomy, thorughly described by Rudebeck and Sundberg (2021), is a two-level classification: error labels are composed of a capital letter, indicating the error's macro-category (Ortography, Lexicon, Morphology, Syntax or Punctuation), followed by a secondary label giving additional information about the type of error and/or the POS involved. The M-Case label, for instance, indicates the presence of a morphological error that has to do with the case inflection of a noun or pronoun.

DaLAJ sentences were parsed with UDPipe 1 (Straka et al., 2016) using the `swedish-talbanken-2.5` model. While not state-of-the-art, UDPipe 1 was preferred over alternatives with higher reported performance due to its speed and ease of use. The resulting dataset is available at `github.com/harisont/L1-L2-DaLAJ`.[13]

4.2.2 Queries

We mentioned the M-Case label, used to mark incorrectly inflected nouns and pronouns. Such a label can be mapped to a rather straightforward single-token query:

```
D1: FEATS_ "Case={$A → $B}"
```

In Swedish, nouns have a definite and an indefinite form. The correct use of these two forms is

[12] The preliminary version of the corpus presented in the paper is exclusively composed of sentences presenting lexical errors. In this work, we used a more recent, soon-to-be-released one also covering morphosyntactical errors.

[13] Last access 05.04.2023.

	Precision	Recall
D_1	77% (76%)	58%
D_2	75%	90%
D_3	89%	62%

Table 2: Precision and recall of the three example queries run on the DaLAJ corpus. Values in parentheses do not take error annotation issues into account.

typically difficult to aquire for L2 learners. As an example of tree query, we therefore use

```
D2: TREE_
      (FEATS_ "Definite={Def → Ind}")
      [AND [DEPREL_ "det",
            FEATS_ "Definite=Def"]]
```

This denotes sentences where a nominal (typically a noun) is in its indefinite form despite being introduced by a definite determiner (typically an article). In terms of SweLL error labels, these cases are a fraction of those marked as M-Def, which is used to indicate a wide variety of errors concerning definiteness (adjective-noun agreement, missing determiners etc.).

Finally, along the lines of the sequence patterns discussed in Section 3.1.2, we use

```
D3: SEQUENCE [
       DEPREL_ "advmod",
        {OR [POS "VERB", POS "AUX"],
         DEPREL_ "nsubj" →
         DEPREL_ "nsubj",
         OR [POS "VERB", POS "AUX"]}]
```

to look for sentences where the V2 order is violated following an adverb or an adverbial clause.[14] With this pattern, we cover some of the errors labelled as S-FinV, namely those involving the misplacement of a finite verb.

4.2.3 Results

As Table 2 suggests, precision is reasonably good even on our automatically annotated data, while recall fluctuates depending on the query.

When it comes to D_1, false negatives are in the almost totality of cases due to the fact that the parser, as it is to be expected, asssigns tokens different dependency labels based on their case (typically, subjects incorrectly inflected in their accusative form are labelled as direct objects and objects in the nominative form become obliques). The vast majority of the false positives is also due not to the query engine itself, but to incorrect alignments deriving from parse errors. In 11 out of 13 such cases, false positives are sentences containing a syntactical error, which seems to confirm the intuition that nonstandard syntax causes the parser to annotate the sentences incorrectly. In only one case a false positive is due to a wrongly assigned error label. More interesting are the cases in which tokens are correctly aligned, but the correction of a syntax error consists in a rephrasing that happens to also alter the case of one of the words involved, such as in the L2 phrase *Rollerna för barn* (literally "The roles for the children"), corrected as *Barnens roller* ("The children's roles"), transforming the nominative *barn* into a genitive *barnens*.

The tree query, more specific, has only 10 expected matches, allowing for a thorough error analysis. The only false negative seems to be due to an alignment error whose cause is hard to pinpoint. Of three false positives, one derives from incorrect morphological annotation, one from a rephrasing that creates problems at the alignment stage and one from the presence, in the L2 sentence *De ligger på första plats i den ligan!* ("They are in first place in that league!"), of the English word "league", annotated (arguably correctly) as an indefinite but translated, in the correction, to the definite *ligan*.

The relatively low recall for D_3 is easily explained by the fact that, as we already discussed, sentences containing syntactical errors are especially challenging for the parser.

5 Conclusions and future work

We presented a query engine for L1-L2 parallel UD treebanks, the first in a larger collection of tools for L2 UD treebanks. The tool, which does not rely on an underlying error taxonomy, allows to search for error-correction pairs via L1-L2 patterns, i.e. pairs of morphosyntactical structures expressed in a pattern matching language for dependency trees, which we extended in order to simplify its use on parallel treebanks. Our novel retrieval algorithm allows searching for full sentences as well as extracting their specific query-matching portions.

Our first, small-scale evaluation of the tool gives promising results, but also shows that the alignment component is often the bottleneck. This propmts us to investigate alignment techniques specifically meant for parallel learner corpora, such as Felice et al. (2016)'s for L2 English. The fact that, for automatically annotated data, many alignment issues derive from parse errors also

[14] Note that, for simplicity, we are only looking for sequences where the verb is contiguous to the subject.

seems to confirm the scarce robustness of standard tools to learner errors, pointing to a need to train *ad hoc* models or explore new, more specific approaches.

In future versions of the tool, we plan to optimize variables and generalize them to all UD fields, thus increasing expressive power of the query language. Furthermore, while the tool is designed with L1-L2 treebanks in mind, nothing prevents us from testing it on multilingual parallel UD treebanks, for example to find instances of known translation divergences.

As for the future of L2-UD, our efforts in the near future will be focused on extracting error patterns from L1-L2 treebanks. Eventually, we hope it will also be possible to integrate the two and enable using error-correction pairs to retrieve similar examples. This would help making the engine more user-friendly, replacing explicit queries, but could also be a strategy to provide L2 learners with feedback, along the lines of Arai et al. (2019).

References

Mio Arai, Masahiro Kaneko, and Mamoru Komachi. 2019. Grammatical-error-aware incorrect example retrieval system for learners of Japanese as a second language. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 296–305, Florence, Italy. Association for Computational Linguistics.

Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. 2016. Universal Dependencies for learner English. *arXiv preprint arXiv:1605.04278*.

Christopher Bryant, Mariano Felice, and Edward Briscoe. 2017. Automatic annotation and evaluation of error types for Grammatical Error Correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada. Association for Computational Linguistics.

Leshem Choshen, Dmitry Nikolaev, Yevgeni Berzak, and Omri Abend. 2020. Classifying syntactic errors in learner language. *arXiv preprint arXiv:2010.11032*.

Elisa Di Nuovo, Manuela Sanguinetti, Alessandro Mazzei, Elisa Corino, and Cristina Bosco. 2022. VALICO-UD: Treebanking an Italian learner corpus in Universal Dependencies. *IJCoL. Italian Journal of Computational Linguistics*, 8(8-1).

Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.

Bruno Guillaume. 2021. Graph matching and graph rewriting: GREW tools for corpus exploration, maintenance and conversion. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–175, Online. Association for Computational Linguistics.

Yan Huang, Akira Murakami, Theodora Alexopoulou, and Anna Korhonen. 2018. Dependency parsing of learner English. *International Journal of Corpus Linguistics*, 23(1):28–54.

Prasanth Kolachina and Aarnte Ranta. 2016. From abstract syntax to Universal Dependencies. In *Linguistic Issues in Language Technology, Volume 13, 2016*.

John Lee, Keying Li, and Herman Leung. 2017a. L1-L2 parallel dependency treebank as learner corpus. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 44–49, Pisa, Italy. Association for Computational Linguistics.

John SY Lee, Herman Leung, and Keying Li. 2017b. Towards Universal Dependencies for learner Chinese. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 67–71, Gothenburg, Sweden.

Juhani Luotolahti, Jenna Kanerva, Sampo Pyysalo, and Filip Ginter. 2015. SETS: Scalable and efficient tree search in dependency graphs. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 51–55, Denver, Colorado. Association for Computational Linguistics.

Scott Martens. 2013. TüNDRA: A web application for treebank search and visualization. In *The Twelfth Workshop on Treebanks and Linguistic Theories (TLT12)*, volume 133, Sofia, Bulgaria.

Arianna Masciolini and Aarne Ranta. 2021. Grammar-based concept alignment for domain-specific Machine Translation. In *Proceedings of the Seventh International Workshop on Controlled Natural Language (CNL 2020/21)*, Amsterdam, Netherlands. Special Interest Group on Controlled Natural Language.

Diane Nicholls. 2003. The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT. In *Proceedings of the Corpus Linguistics 2003 conference*, volume 16, pages 572–581, Lancaster, UK. Cambridge University Press Cambridge.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. *arXiv preprint arXiv:2004.10643*.

Petr Pajas and Jan Štěpánek. 2009. System for querying syntactically annotated corpora. In *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 33–36, Suntec, Singapore. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Aarne Ranta and Prasanth Kolachina. 2017. From Universal Dependencies to abstract syntax. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 107–116, Gothenburg, Sweden. Association for Computational Linguistics.

Lisa Rudebeck and Gunlög Sundberg. 2021. SweLL correction annotation guidelines. In *The SweLL guideline series nr 4*, Gothenburg, Sweden. Institutionen för svenska, Göteborgs universitet.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Elena Volodina, Lena Granstedt, Arild Matsson, Beáta Megyesi, Ildikó Pilán, Julia Prentice, Dan Rosén, Lisa Rudebeck, Carl-Johan Schenström, Gunlög Sundberg, et al. 2019. The SweLL language learner corpus: From design to annotation. *Northern European Journal of Language Technology*, 6:67–104.

Elena Volodina, Yousuf Ali Mohammed, and Julia Klezl. 2021. DaLAJ-a dataset for linguistic acceptability judgments for Swedish: Format, baseline, sharing. *arXiv preprint arXiv:2105.06681*.

Appendix A Abbreviations

A.1 UD standard

A.2 CoNNL-U fields

- `DEPREL`: dependency label;
- `FEATS`: list of morphological features;
- `FORM`: word form or punctuation symbol;
- `LEMMA`: lemma/stem of the word form;
- `POS`: Universal POS tag;
- `XPOS`: language-specific POS tag.

For the full specification of the CoNNL-U format, see `universaldependencies.org/format.html` (last access 05.04.2023).

A.2.1 Universal POS tags

- `ADP`: adposition (pre- or postposition);
- `ADV`: adverb;
- `AUX`: auxiliary;
- `DET`: determiner;
- `PRON`: pronoun;
- `VERB`: non-auxiliary verb.

For a comprehensive list of UD POS tags, see `universaldependencies.org/u/pos` (last access 05.04.2023).

A.2.2 Universal dependency relations

- `advmod`: adverbial modifier of a predicate or modifier word;
- `amod`: adjectival modifier of a nominal;
- `aux`: auxiliary;
- `case`: case-marking element treated as a separate word;
- `det`: determiner. The subtype `poss` indicates a possessive;
- `nsubj`: nominal subject;
- `obj`: direct object;
- `obl`: oblique nominal, i.e. non-core verb argument or adjunct;
- `root`: root of the sentence, usually its main (non-auxiliary) verb and, in general, a content word.

For a comprehensive list of UD relations, see `universaldependencies.org/u/dep` (last access 05.04.2023).

A.3 VALICO-UD error labels

- IDG: Determiner incorrectly Inflected for Gender. The IDGcascade label is used in cases where the incorrect inflection depends on another error, typically ING or RN;
- ING: Noun incorrectly Inflected for Gender;
- RD: Determiner Replacement (wrong choice of determiner);
- RN: Noun Replacement (lexical error involving a noun);
- SEU: Spelling error - Unnecessary apostrophE.

The complete error annotation guidelines for the VALICO-UD treebank are available at `raw.githubusercontent.`

com/ElisaDiNuovo/VALICO-UD_guidelines/main/Error_Annotation_Guidelines_v.1.1.pdf (last access 05.04.2023).

A.4 SweLL error labels

- M-Case: noun or pronoun incorrectly inflected for case;
- M-Def: definiteness-related error, namely either:
  - noun, pronoun, adjective or participle incorrectly inflected for definiteness;
  - incorrect, missing or redundant article;
- S-FinV: incorrect placement of a finite verb.

See Rudebeck and Sundberg (2021) for the full SweLL correction annotation guidelines.

Appendix B Annotation inconsistencies

B.1 VALICO-UD corpus

- Sentence 34-12_en-3: *{un → un'*}* correctly labelled as SEU, but not as IDG[15]
- sentence 19-06_en-3 *un* in *{un uomo → un'uoma*}* labelled as IDG rather than IDG-cascade
- sentence 18-10_en-1 *la* in *{sul braccio → sul+la braccia*}* labelled as IDG rather than IDGcascade
- sentence 18-05_en-1 token *sedile* lacking gender annotation in both the L1 and the L2 files
- sentece 17-07_en-2 token *amante* lacking gender annotation in both the L1 and the L2 files
- sentece 3-13_fr-3 *le* in *{gli uccelli → le uccele*}* labelled as IDG rather than IDGcascade.

This list of error annotation issues refers to the 14.05.2022 version of the treebank.[16] At the time of writing, these observations have been discussed with the authors of the treebank and the annotations in question, excepts for two cases in which they were the result of a deliberate choice, are in the process of being fixed.

B.2 DaLAJ corpus

- Sentence 1811 labelled as S-Clause rather than M-Case.

This annotation error refers to the 04.02.2023 version of the treebank and was corrected in a subsequent update.[17]

[15] The sentence in question is *Ho voltato la pagina e ho iniziato a leggere {un → un'*} altro titolo.* ("I turned the page and started reading another title."). This is an interesting case: the UD annotation correctly states that the indefinite article *un* is masculine while its L2 counterpart *un'* (mind the apostrophe) is feminine, but the manually assigned error tag, rather than the expected IDG, is SEU, which indicates, also correctly, a spelling error (unnecessary apostrophe). All the more reasons not to rely on explicit error labelling!

[16] Available for download at github.com/UniversalDependencies/UD_Italian-Valico (last access 05.04.2023), with commit SHA 7c4fae4f1e6491ca9e648cfb902e1c675c179a42.

[17] Available for download at github.com/harisont/L1-L2-DaLAJ (last access 05.04.2023), with commit SHA 94e133aa083e487cfb28a7c22dda4e1c240bcaf5.

Appendix C Example program output: Markdown reports

The following reports, as well as all results presented in this paper, were obtained with L2-UD v0.[18]

C.1 `TREE_ (AND [POS "NOUN", FEATS_ "Gender=$A"]) [AND [POS "DET", FEATS_ "Gender=$A → $B"]]` (V_1')

Sentence 30-09_de-2:

L1 sentece	L2 sentece
Poi , lei si è arrabbiata e mi ha detto che questo uomo con i grandi muscoli che si è sdraiato a terra era il suo fidanzato e **il suo** grande **amore** .	Poi , lei si è arrabbiata e mi ha detto che questo uomo con i grandi muscoli che si è sdraiato sulla terra era il suo fidanzato e **la sua** grande **amore** .

Sentence 3-11_fr-3:

L1 sentece	L2 sentece
La donna ringraziava il suo salvatore con un abbraccio e chiudeva **gli occhi** .	La dona ringraziava suo salvatore con un braccio e chiusa **le occhi** .

Sentence 10-07_es-1:

L1 sentece	L2 sentece
Gli ho gridato **alcune parolacce** .	L’ ho gridato **alquini parolace** .

Sentence 4-04_fr-2:

L1 sentece	L2 sentece
Un **altro uomo** si trovava lì , seduto su una panchina del di il parco , leggendo un giornale con i suoi occhiali .	Un **altra uomo** , si trova li , seduto sul su il un panchino del di il parco , leggendo un giornale con i suoi occhiali .

Sentence 34-12_en-3:

L1 sentece	L2 sentece
Ho voltato la pagina e ho iniziato a leggere **un** altro **titolo** .	Ho voltato la pagina e ho iniziato a leggere **un’** altro **titolo** .

Sentence 19-01_en-3:

L1 sentece	L2 sentece
Ieri al a il parco , **un uomo** brutto è arrivato e ha detto delle parole cattive a una donna .	Ieri al a il parco , **un’ uomo** brutto è arrivata e ha detto le parole cattive a una donna .

Sentence 27-02_de-3:

[18] Download link: `github.com/harisont/L2-UD/releases/tag/v0` (last access 05.04.2023).

L1 sentece	L2 sentece
Subito guarda come un altro uomo con grande forza fisica e con malumore porta sulle **sue spalle** una ragazza che grida .	Subito guarda come un altro uomo con grande forza fisica e con malumore porta una ragazza sulle **sui spalle** che grida .

Sentence 3-12_fr-3:

L1 sentece	L2 sentece
Era **un** vero **momento** di benessere .	Era **una** vero **momento** di benessere .

Sentence 27-08_de-3:

L1 sentece	L2 sentece
« Il **mio amore** non dipende dal suo comportamento . »	« Il **mia amore** non dipende dal suo comportamento . »

Sentence 32-06_de-2:

L1 sentece	L2 sentece
Un uomo molto forte , intelligente , sportivo e carino mi ha salvato da **questa** ignorante **persona** , Marco . **Un uomo** molto forte , intelligente , sportivo e carino mi ha salvato da questa ignorante persona , Marco .	Un' uomo molto forte , intelligente , sportivo e carino mi ha salvato di **questo** ignorante **persona** Marco . **Un' uomo** molto forte , intelligente , sportivo e carino mi ha salvato di questo ignorante persona Marco .

C.2 `SEQUENCE [DEPREL_ "advmod", OR [POS "VERB", POS "AUX"], DEPREL_ "nsubj" -> DEPREL_ "nsubj", OR [POS "VERB", POS "AUX"]]` (D_3)

Sentence 1958:

L1 sentece	L2 sentece
Därför tycker jag om havet .	**Därför jag tycker** om havet .

Sentence 1943:

L1 sentece	L2 sentece
Tyvärr har någonting hänt som gör att jag inte kan gå på kursen och jag önskar att få pengarna tillbaka .	**Tyvärr någonting har hänt** som gör att jag inte kan gå på kursen och jag önskar att få pengarna tillbaka .

Sentence 1950:

L1 sentece	L2 sentece
Därför vill jag inte **flytta** .	**Därför jag vill** inte **flytta** .

Sentence 1965:

L1 sentece	L2 sentece
Där bodde jag i Göteborg med min mamma och hennes hundar i hennes hus .	**Där jag bodde** i Göteborg med min mamma och hennes hundar i hennes hus .

Sentence 1936:

L1 sentece	L2 sentece
Ibland **kan vi titta** och lyssna på hur det funkar .	Ibland **vi kan titta** och lyssna på hur det funkar .

Sentence 1139:

L1 sentece	L2 sentece
Därför har **man kommit** med förslaget att ha en kurs angående arbetslivet i gymnasiet .	**Därför man kommit** med förslaget att ha en kurs angående arbetslivet i gymnasiet .

Sentence 1942:

L1 sentece	L2 sentece
Lyckligtvis kan du spela piano , och **där** kan **du lära** känna nya vänner .	Lyckligtvis kan du spela piano , och **där du** kan **lära** känna nya vänner .

Sentence 1976:

L1 sentece	L2 sentece
Ibland brukar man säga att kärleken har ingen gräns och det är sant .	**Ibland man brukar** säga att kärleken har ingen gräns och det är sant .

Sentence 1939:

L1 sentece	L2 sentece
Men nu **är jag** inte **intresserad** av den längre .	Men nu **jag är** inte **intresserad** av den längre .

Filtering Matters: Experiments in Filtering Training Sets for Machine Translation

Steinþór Steingrímsson[1], **Hrafn Loftsson**[1], **and Andy Way**[2]
[1]Department of Computer Science, Reykjavik University, Iceland
[2]ADAPT Centre, School of Computing, Dublin City University, Ireland
steinthor18@ru.is, hrafn@ru.is, andy.way@adaptcentre.ie

Abstract

We explore different approaches for filtering parallel data for MT training, whether the same filtering approaches suit different datasets, and if separate filters should be applied to a dataset depending on the translation direction. We evaluate the results of different approaches, both manually and on a downstream NMT task. We find that, first, it is beneficial to inspect how well different filtering approaches suit different datasets and, second, that while MT systems trained on data prepared using different filters do not differ substantially in quality, there is indeed a statistically significant difference. Finally, we find that the same training sets do not seem to suit different translation directions.

1 Introduction

In recent years, machine learning (ML) research has generally focused on creating better models rather than better datasets. The focus on benchmarking model performance spurs researchers into adapting the largest existing datasets without fully considering fidelity to the underlying problem the model should solve (Mazumder et al., 2022). The effectiveness of ML models, however, depends on both algorithms and data. Aroyo et al. (2022) argue that as the datasets define the world within which models exist and operate, more work is needed on how the data can be optimized for more effective use.

Filtering parallel data for machine translation (MT) is the task of removing possible detrimental segments from data used for training MT models. Detrimental segments, in this context, are sentence pairs in the training data that may degrade the performance of an MT system trained on the data. Filtering is usually carried out using a set of rules, scoring mechanisms and/or classifiers, to remove sentence pairs with the lowest perceived quality.

In this work, we experiment with filtering the raw data from two parallel corpora, ParaCrawl (Bañón et al., 2020) and ParIce (Barkarson and Steingrímsson, 2019), for the English–Icelandic language pair. Our goal is to minimize detrimental data while losing little or no useful data from the texts, thus building a more accurate training set.

We investigate how shallow filters, four different scoring mechanisms and different classifiers based on them are suited to score and filter English–Icelandic sentence pairs. We compare these to Bicleaner (Sánchez-Cartagena et al., 2018; Ramírez-Sánchez et al., 2020) and to how the two corpora (ParaCrawl and ParIce) were filtered for publishing.

Recent literature on parallel corpus filtering has largely focused on filtering noisy data collected from the web, as discussed in Section 2. We want to investigate whether the same approaches are suitable to filter noisy web-crawled corpora and cleaner parallel corpora compiled from document pairs that are known to be mutual translations. Furthermore, although the same training data is usually used for training both translation directions, source→target and target→source, for a given dataset, we investigate whether that is optimal or whether filtering separately for each translation direction is likely to bring improvements to a downstream MT task.

Our primary goal is to find out how to filter parallel corpora so as to compile a training set that potentially gives the best results when used to train an MT system. We seek an answer to the following research questions: **1) Should the same filtering approaches be used for a given language pair, regardless of the datasets being filtered? 2) Should the intended translation di-**

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 588–600

rection of an MT system effect how the data, used to train the system, is filtered? In order to answer these questions, we build MT models for both translation directions and multiple different filtering approaches for each one, and evaluate the results, both manually and automatically. We find for best results, specific filtering approaches should be chosen based on the dataset and translation direction being filtered.

2 Related Work

In their paper, Khayrallah and Koehn (2018) show that incorrect translations, untranslated target text, misalignments, and other noisy segments in a parallel corpus have a detrimental effect on the output quality of NMT systems trained on that corpus, as measured using BLEU (Papineni et al., 2002). They specify five general classes of noise commonly found in the German-English ParaCrawl corpus: misaligned sentences, disfluent text, wrong language, short segments, and untranslated sentences. As this classification is rather coarse, some variation can be expected within each class; a misalignment in one sentence pair does not have to be equivalent to a misalignment in another sentence pair.

Briakou and Carpuat (2021) focus on fine-grained semantic divergences within mostly equivalent pairs (pairs of words, phrases or sentences that have similar meanings and connotations). An example given in the paper is fr: "votre père est français" → en: "your parent is french", where the correct translation should be: "your father is french". These fine-grained divergences can even be found in high-quality parallel corpora. They find that the divergences cause degradation on the MT output of a system trained on the data, as measured by BLEU and METEOR (Banerjee and Lavie, 2005), and that divergences impact model confidence in their predictions. Lexical substitution causes the largest degradation and subtree deletion the least. Nevertheless, the impact on divergences seem to be smaller than that of noise. They argue that this suggests that noise-filtering techniques are subobtimal to deal with fine-grained divergences.

In early work on filtering web-scraped parallel corpora, Rarrick et al. (2011) filter out machine-translated content and show that removing large amounts of training data can improve performance of an MT system, challenging conventional wisdom at the time that more data is better.

Cross-lingual word embeddings have been used to calculate distance between equivalences in different languages (Luong et al., 2015; Artetxe et al., 2016). Defauw et al. (2019) treat filtering as a supervised regression problem and show that Levenshtein distance (Levenshtein, 1966) between the target and MT-translated source, as well as cosine distance between sentence embeddings of the source and target, are important features.

The Conference on Machine Translation, WMT, hosted three annual shared tasks on parallel corpus filtering (Koehn et al., 2018, 2019, 2020), focusing on filtering noisy web-crawled corpora. Chaudhary et al. (2019) and Artetxe and Schwenk (2019a) introduced approaches based on cross-lingual sentence embeddings trained from parallel sentences. When using cosine similarity to find the nearest neighbours in an embedding space, cosine similarity is not necessarily globally consistent and different scales of target candidates for a given source sentence may affect their relative ranking, causing the hubness problem, described by Dinu and Baroni (2015). The problem is caused by a few vectors in the embedding space that tend to be "universal" neighbours, i.e., neighbours of a large number of different mapped vectors, pushing the correct ones down the neighbour list. Both papers tackle the scale inconsistencies of cosine similarity by considering the margin between a given sentence pair and its closest candidates to normalize the similarity scores.

Bicleaner uses a set of handcrafted rules to detect flawed sentences and then proceeds to use a random forest classifier based on lexical translations and several shallow features such as respective length, matching numbers and punctuation. It also scores sentences based on fluency using 5-gram language models. Bicleaner AI (Zaragoza-Bernabeu et al., 2022) is a fork of Bicleaner using a neural classifier. It has been shown to give significant improvements in translation quality as measured by BLEU when used for filtering training data for multiple language pairs, as compared to the previous version of the tool. In contrast to tools that implement a single method for parallel corpus filtering, Aulamo et al. (2020) implement multiple different filters in the OpusFilter toolbox. These include heuristic based filters, language identification, character-based language models and word alignment tools.

Herold et al. (2022) revisit the noise classes specified by Khayrallah and Koehn (2018) to investigate how accurately two of the strongest filtering approaches to date (according to them) cross entropy (Rossenbach et al., 2018) and LASER (Artetxe and Schwenk, 2019b) can filter out different noise classes. They find that for a common language pair, German→English, most types of noise can be detected with over 90% accuracy, although misalignments and poor synthetic translation can only be detected with an accuracy of less than 70%. For a less common language pair, Khmer–English, the performance of the filtering system degraded significantly and the accuracy of identifying noise was lowered by 8–19%, depending on the type of noise.

3 Experimental Setup

We compare a number of approaches and scoring mechanisms and apply them to a web-crawled corpus, on the one hand, and a parallel corpus compiled from parallel documents, on the other. We manually evaluate samples of the results using the taxonomy developed by Kreutzer et al. (2022) to gain an understanding of what sort of data each approach and scoring mechanism filters out. We then train MT systems on datasets filtered using the different approaches, as well as on previously published, filtered versions of the corpora, and compare the quality of the resulting systems in terms of BLEU scores. We measure BLEU scores on the test set provided for the English–Icelandic language pair in the WMT 2021 shared task (Akhbardeh et al., 2021), using SacreBLEU (Post, 2018).

3.1 Data Sets

The data sets we use for our experiments are the English–Icelandic part of ParaCrawl and the English–Icelandic parallel corpus ParIce. We carry out the same experiments using both corpora and compare the results.

ParaCrawl is compiled from web-crawled data. Based on the evaluation by Kreutzer et al. (2022), approximately 76% of sentence pairs are acceptable mutual translations, on average, in 21 language pairs from the ParaCrawl 7.1 datasets cleaned for publication. There is also high variance between languages and low-resource datasets tend to have lowest human-judged quality. Rikters (2018) inspects the quality of the first version of ParaCrawl and filters out 85% of the English–Estonian ParaCrawl dataset. Although there may be differences in noise ratio between different versions of the corpus, for most language pairs ParaCrawl can likely be made more useful for training MT models by better filtering. This has been emphasized by the results of the WMT shared tasks on filtering parallel corpora. In our work, we start with the raw data from version 9 of the corpus, consisting of 65,373,727 sentence pairs in total. Our goal is to extract from the corpus sentence pairs useful for training MT systems on its own or to complement other data sets, and leave out sentence pairs likely to be detrimental.

The English-Icelandic parallel corpus ParIce differs from ParaCrawl in that it is compiled from known parallel documents, which have been aligned at the sentence level. When the corpus was compiled initially, the filtering process resulted in an estimated 20% reduction in corpus size. Out of what remained, manual evaluation of samples from the corpus indicated that approximately 3.5% was in some way faulty (Barkarson and Steingrímsson, 2019). The corpus is available unfiltered, accompanied with semantic similarity scores for each sentence pair and flags indicating whether it is recommended to filter out the pair or not. We work with the unfiltered data, version 21.10 (Steingrímsson and Barkarson, 2021).

3.2 Filters and Scoring

In order to find which sentence pairs are useful and which ones to filter out, we use an array of tools for scoring sentence pairs to find the highest-quality data within the corpora. We start with shallow filters to remove pairs that are very likely to be noise, and then proceed to run different tools, both made available by others and of our own device.

Shallow Filters

Our shallow filters are inspired by Pinnis (2018), who applies 17 different filters in his work. We do not use all his filtering approaches but select the ones likely to remove the highest portion of detrimental pairs as outlined by Khayrallah and Koehn (2018). They are:

1. If both source and target sentence have 3 tokens or less, the pair is discarded.
2. All pairs, for which 60% or more of the tokens in one language are also present in the other language, are removed.

3. At minimum, 70% of characters in both sentences should be alphabetical.

4. Both languages are in the top 2 prediction of a language filter. We use fasttext (Joulin et al., 2017) for language filtering.

5. Removal of near-duplicate pairs. We consider sentence pairs with all non-alphabetical symbols removed and if there are identical pairs in the corpus we keep the one with the highest score (Bicleaner score for ParaCrawl, LASER, LaBSE and WAScore for ParIce).

6. Removal of near-duplicate source or target sentence. We consider strings after removing all non-alphabetical symbols, and all tokens starting with a capital letter (removing possible named entities) from the sentences. If there are identical such strings in the same language, we select the highest scoring pair.

Bicleaner Models

Bicleaner is an open source noise filter and classification tool to clean parallel corpora, released as part of the ParaCrawl project and used to generate the filtered ParaCrawl datasets. Bicleaner uses a set of hard rules for pre-filtering, n-gram models for fluency scoring, and a random forest classifier to produce a probability score using features such as lexical similarity, sentence length, punctuation and capitalization. Bicleaner AI is a fork that uses a fine-tuned XLM-RoBERTa classifier to produce probability scores by training it on positive samples from existing parallel corpora and negative samples which are created by corrupting the same positive samples. In synthesising the noise, the tool tries to emulate errors commonly introduced by sentence segmentation and alignment.

We use two publicly available Bicleaner models, version 1.5, for English–Icelandic, and Bicleaner AI 1.0 full model. In addition, we train two new models using Bicleaner v0.15.2, one that classifies lemmatized data and the other unlemmatized. For training each model, we used word frequency information from the Icelandic Gigaword Corpus (Steingrímsson et al., 2018) for Icelandic and News Crawl (Barrault et al., 2020) for English, a probabilistic dictionary (Steingrímsson et al., 2022), and for parallel training data, 250k highest-scoring sentence pairs from the 21.10 version of ParIce (Steingrímsson and Barkarson, 2021), based on the scores published with the corpus.

Scoring and Score-Based Classifiers

We use multiple scoring mechanisms to assess the quality of the bilingual sentence pairs.

LASER (Artetxe and Schwenk, 2019b) uses a single BiLSTM encoder with a shared byte-pair encoding (BPE) vocabulary (Sennrich et al., 2016) for all languages and is trained on parallel corpora.

LaBSE (Feng et al., 2022) is trained and optimized to produce similar representations for bilingual sentence pairs. It uses dual encoder models, with the encoder architecture following the BERT Base model, and additive margin softmax which extends the scoring function in the model by introducing a large margin around positive pairs, improving the separation between translations and nearby non-translations (Yang et al., 2019). An available pre-trained model was trained on 109 languages, including Icelandic and English.

NMTScore (Vamvas and Sennrich, 2022) is based on translation cross-likelihood, the likelihood that a translation of segment *A* into some language, could also be a translation of segment *B* into the same language. An example could be the translation of the French 'Bonjour!' into the Swedish 'Hej!'. To calculate translation cross-likelihood, the French segment would first be translated to a third language, say English, and the score is based on the probability of the model getting the same translation for the Swedish segment. The score is symmetrized by averaging the translation probabilities in both directions. We use the M2M100 multilingual translation model (Fan et al., 2021) to calculate NMTScore.

WAScore is a word alignment-based score devised to measure word-level parallelism, introduced in Steingrímsson et al. (2021) to help with identifying parallel bilingual sentence pairs in a comparable corpus.

The scores are used to train classifiers for determining acceptability of parallel sentence pairs. We adapt a training set compiled for a classifier used in mining comparable corpora (Steingrímsson et al., 2021). The dataset was compiled of 50,000 randomly sampled non-parallel pairs from two monolingual news corpora for negative examples and 1,000 parallel segments containing sentence pairs from news media. LASER, LaBSE, NMTScore and WAScore were calculated for all 51K sentence pairs, and used to train the classifiers. We used scikit-learn (Pedregosa et al., 2011) to train random forest (Breiman, 2001), support

ParaCrawl shallow filtering					
Filter	Dataset Size	CC (%)	3C (%)	X (%)	3X (%)
0. ParaCrawl v9 Raw	65,373,727	14.40	69.20	8.00	30.80
0b. ParaCrawl v9 Clean	2,967,519	13.60	89.20	8.80	10.80
1.-3. Non-zero / low overlap (accepted)	31,094,385	23.60	94.80	4.40	5.20
1.-3. Non-zero / low overlap (discarded)	34,285,591	1.60	46.80	9.20	53.20
4.-5. Symbol+Language filter (accepted)	26,609,214	25.00	97.20	2.80	2.80
4.-5. Symbol+Language filter (discarded)	4,485,171	11.20	85.60	9.20	14.40
6. Similar pairs (accepted)	4,666,464	12.00	86.80	12.80	13.20
7. Similar segments (accepted)	2,081,354	14.80	95.60	3.60	4.40
ParIce shallow filtering					
0. ParIce 21.10 filtered	1,776,049	73.60	95.20	4.80	4.80

Table 1: Size and manual evaluation results for the shallow filtering approaches. For each dataset 250 randomly sampled pairs are evaluated. 3C stands for all correct codes: CC, CB and CS. 3X stands for all error codes: X, WL and NL. For comparison, we also evaluate the clean version of the corpus as published by the ParaCrawl project. Note that we evaluated both accepted and discarded pairs for two of the filtering steps.

vector machine (Cortes and Vapnik, 1995) and logistic regression (Cox, 1958) classifiers.

Sentence Perplexity using GPT-2

Manual evaluation of ParaCrawl sentence pairs revealed that the Icelandic sentences in ParaCrawl are frequently ungrammatical or have erratic syntax, even though some, and in some cases most or all, of the lexical semantics of the translations are correct. This is likely because many web pages, scraped by the ParaCrawl project, use MT models to generate texts in multiple languages, even though the MT models do not generate fluent results. We try to find these badly formed sentences by training a classifier to recognize fluent and disfluent sentences. The classifier uses a pre-trained GPT-2 model (Radford et al., 2019), trained on the Icelandic Gigaword Corpus (Steingrímsson et al., 2018).[1] To train the classifier, we selected 10,000 sentences randomly from WikiMatrix (Schwenk et al., 2021) and ParaCrawl v8, and manually classified them in two groups: *coherent* (6,570 sentences) and *incoherent* (3,430 sentences).[2] The classifier uses the GPT-2 model to calculate perplexity for the sentences, and chooses potential thresholds as the average between two adjacent perplexity values. It then uses a maximization function to decide on a threshold that yields the most accurate prediction based on the training set.

[1]The model, trained by Jón Friðrik Daðason, is available on Hugging Face: `https://huggingface.co/jonfd/gpt2-igc-is/tree/v1.0`.

[2]Dataset available here: `https://github.com/steinst/filter-align-datasets`

3.3 Manual Evaluation

We manually annotated samples of the data sets compiled by each filtering approach. In our evaluation, we followed the taxonomy developed by Kreutzer et al. (2022), but slightly amended one category, CB, to include partial alignments. The taxonomy uses three codes for correct pairs and three error codes:

- CC – Correct translation, natural sentence.
- CB – Correct translation, boilerplate, partial alignments or grammatical errors.
- CS – Correct translation, short.
- X – Incorrect translation.
- WL – Either sentence in wrong language.
- NL – Either sentence is non-linguistic content.

Shallow Filters: We annotated 250 randomly selected pairs from the datasets at different stages of shallow filtering. Table 1 shows the size of the datasets after applying shallow filters, and the percentage of sentence pairs in different categories. The evaluation indicates that almost 70% of the raw ParaCrawl data is potentially useful, while over 30% is in the best case useless and possibly detrimental. Note that this describes the dataset before any filtering or deduplication has been carried out. ParaCrawl also distributes a cleaned version of the corpus, containing approximately $3M$ sentence pairs. In that version, over 10% of sentence pairs are still erroneous and, while almost

	Laser								LaBSE							
	ParaCrawl				ParIce				ParaCrawl				ParIce			
	CC	3C	X	3X	CC	3C	X	3X	CC	3C	X	3X	CC	3C	X	3X
0.0 ⇒ 0.1	10	100	0	0	95	100	0	0	0	7	93	93	1	9	91	91
0.1 ⇒ 0.2	9	99	1	1	93	99	1	1	0	5	95	95	4	12	88	88
0.2 ⇒ 0.3	8	99	1	1	92	100	0	0	1	6	94	94	11	26	74	74
0.3 ⇒ 0.4	16	100	0	0	87	100	0	0	0	7	93	93	14	50	50	50
0.4 ⇒ 0.5	16	99	1	1	83	99	1	1	2	13	85	87	24	75	25	25
0.5 ⇒ 0.6	20	85	14	15	75	98	2	2	4	42	57	58	46	93	7	7
0.6 ⇒ 0.7	15	69	31	31	61	90	10	10	16	71	29	29	64	98	2	2
0.7 ⇒ 0.8	10	43	57	57	58	93	7	7	26	94	6	6	82	100	0	0
0.8 ⇒ 0.9	13	56	42	44	63	75	25	25	15	98	2	2	89	99	1	1
0.9 ⇒ 1.0	27	63	36	37	51	65	36	36	11	99	1	1	99	100	0	0
	NMTScore								WAScore							
	ParaCrawl				ParIce				ParaCrawl				ParIce			
	CC	3C	X	3X	CC	3C	X	3X	CC	3C	X	3X	CC	3C	X	3X
0.0 ⇒ 0.1	22	76	24	24	65	92	8	8	1	17	81	83	8	45	55	55
0.1 ⇒ 0.2	20	96	4	4	87	100	0	0	12	46	53	54	43	91	9	9
0.2 ⇒ 0.3	12	98	2	2	85	100	0	0	28	72	21	28	57	95	5	5
0.3 ⇒ 0.4	9	100	0	0	94	100	0	0	27	88	9	12	73	97	3	3
0.4 ⇒ 0.5	9	100	0	0	97	100	0	0	39	96	4	4	80	100	0	0
0.5 ⇒ 0.6	12	99	1	1	97	100	0	0	33	95	5	5	92	100	0	0
0.6 ⇒ 0.7	13	100	0	0	93	100	0	0	27	93	7	7	93	99	1	1
0.7 ⇒ 0.8	11	99	0	1	99	100	0	0	10	99	1	1	94	99	1	1
0.8 ⇒ 0.9	23	100	0	0	100	100	0	0	7	97	3	3	94	99	1	1
0.9 ⇒ 1.0	20	100	0	0	100	100	0	0	5	98	2	2	95	100	0	0

Table 2: Results of the manual evaluation of samples of 100 randomly selected sentence pairs from each of ten bands for the scoring mechanisms used.

90% are potentially useful, only 13.6% are evaluated to be good mutual translations. We filter the raw data and evaluate the changes after in between shallow filtering steps. All the filters discard some mutual translations but proportionally more inadequate pairs. While the 3C column indicates the ratio of all pairs in the correct category, it includes boilerplate and ungrammatical segments not necessarily useful for MT. We want our filters to keep as many sentence pairs from the CC category and remove all from the X-categories. After filters 1-7 (see Section 3.2) have been applied, we see the number of pairs annotated as correct, CC, is 14.8%. After filter 5 this was 25%, but the last two filters lower the ratio because sentences that are identical, except for numbers or other named entities, have been reduced to one example.

We only manually evaluate the ParIce corpus after applying all the shallow filters and do not investigate the changes at each stage. This is because the data in the corpus all comes from known document sources and should not contain as much noisy data as ParaCrawl. We find that about 5% of sentence pairs in the corpus are erroneous, a number largely in line with the original ParIce paper, where the evaluation indicated that 3.5% of the alignments were bad, but we also find that only about three out of every four sentence pairs are mutual translations, with about 20% being accepted as correct while being imperfect in some way, usually due to misalignments.

Scores: After evaluating the shallow filters, we evaluated the effectiveness of the scoring mechanisms. We divided the scores for each scoring approach into 10 bands, and manually evaluated 100 pairs for each band. The evaluation results, shown in Table 2, indicate that all the scoring methods have some merit and could probably be useful to a classifier. On their own, the results usually differ depending on the parallel corpora used, with the accuracy of the same scoring mechanism varying for different corpora. For example, the LaBSE score has to be more than 0.7 for more than 90% of sentence pairs in a scoring band to be acceptable (3C) for ParaCrawl, but only 0.5 for ParIce.

The distribution of the scores differ between scoring approaches, which can effect their usefulness. While NMTScore seems to be very accurate when looking at the bands in the table, 83% of the ParaCrawl sentences have a score of less than 0.3, and 25% of the ParIce sentences have a score of less than 0.1, indicating that even though the results seem very good, using only this scoring method may not be enough for accurate filtering. It should also be noted that most approaches do not seem to be very good at discerning finer nuances such as whether a sentence pair contains only mutual translations or if there is additional content in at least one of the sentences. The ratio of CC usually does not change as consistently with rising scores as the 3C or 3X ratio. This may indicate that if some sentence pairs classified as CB are detrimental to MT training, we need other approaches to identify them and filter out.

Filters: We annotated 100 pairs from each group of stochastic filtering approaches. We use the clas-

Filter	ParaCrawl Filters									
	accepted (%)					rejected (%)				
	No. pairs	CC	3C	X	3X	No. pairs	CC	3C	X	3X
GPT-2	1,218,256	15	93	7	7	863.098	5	91	8	9
Logistic Regression	1,940,385	38	85	4	15	140,969	18	37	61	63
Random Forest	1,981,405	7	98	0	2	99,949	2	22	77	78
Support Vector Machine	1,991,924	12	98	2	2	89,430	0	22	78	78
Bicleaner baseline (0.50)	1,973,885	22	96	4	4	107,469	10	41	58	59
Bicleaner baseline (0.67)	1,705,042	15	98	2	2	376,312	20	80	20	20
Bicleaner retrained (0.50)	1,898,209	25	97	3	3	183,145	27	75	25	25
Bicleaner retrained (0.67)	1,615,913	20	98	2	2	465,441	24	81	18	19
Bicleaner lemmatized (0.50)	1,850,884	18	88	8	12	230,470	14	66	28	34
Bicleaner lemmatized (0.67)	1,512,437	30	93	5	7	568,917	21	70	29	30
Bicleaner AI (0.50)	1,235,771	**33**	**99**	1	1	845,583	6	85	13	15
Bicleaner AI (0.67)	1,096,288	25	97	3	3	985,066	8	92	8	8
Filter	**ParIce filters**									
	accepted (%)					rejected (%)				
	No. pairs	CC	3C	X	3X	No. pairs	CC	3C	X	3X
GPT-2	1,444,956	81	96	4	4	331,093	68	91	9	9
Logistic Regression	1,560,346	85	100	0	0	215,703	49	77	23	23
Random Forest	1,667,847	86	99	1	1	108,202	20	51	49	49
Support Vector Machine	1,646,183	91	100	0	0	129,866	28	58	42	42
Bicleaner baseline (0.50)	1,546,216	85	99	1	1	229,833	35	79	21	21
Bicleaner baseline (0.67)	1,242,258	86	100	0	0	533,791	48	86	14	14
Bicleaner retrained (0.50)	1,499,610	85	99	1	1	276,439	42	90	10	10
Bicleaner retrained (0.67)	1,244,412	94	100	0	0	531,637	55	95	5	5
Bicleaner lemmatized (0.50)	1,463,780	89	100	0	0	312,267	50	90	10	10
Bicleaner lemmatized (0.67)	1,117,814	88	100	0	0	604,235	69	98	2	2
Bicleaner AI (0.50)	1,262,313	**95**	**100**	0	0	513,736	60	86	13	14
Bicleaner AI (0.67)	1,152,319	91	100	0	0	623,730	77	95	5	5

Table 3: Manual evaluation of datasets generated by different filtering approaches. We both evaluate sentence pairs accepted by each filtering approach, and rejected by it.

sifiers and Bicleaner models described in section 3.2. We set cutoff score at two different levels for each Bicleaner model, 0.5 and a higher threshold of 0.67 to try to discover whether detrimental sentence pairs can still be found at such a high level.

As evident in Table 3, the filtering mechanisms are quite adept at removing erroneous sentence pairs. We can see that for both corpora, all but two filters return over 90% accepted sentence pairs, and a low rate of erroneous data, and for ParIce, in particular, almost all erroneous data is removed for 8 out of 12 filtering approaches. However, as it is important to keep as many of the good sentence pairs as possible, the most useful approaches may be the ones that remove the fewest mutual translations. We see that while the Bicleaner AI model has the highest proportion of CC, mutual translations, it has the drawback of filtering out the highest proportion of sentences compared to almost all other approaches. Almost half of the ParaCrawl data was rejected, 985,066 out of 2,081,354 sentence pairs, when the threshold score is set to 0.67, of which 92% were rated in one of the correct categories. In order to investigate further what is best for MT training, we next train multiple models, using all the different data sets we have compiled, in order to see how the translations generated by these models compare to the results of our manual evaluation.

3.4 Automatic evaluation

We evaluate the effect of different filtering approaches on a downstream NMT task by training different MT models for each of the compiled datasets and evaluate them using BLEU. We use fairseq (Ott et al., 2019) to train Transformer$_{\mathrm{BASE}}$ models, as described in Vaswani et al. (2017), except that we set dropout to 0.2 and use BPE with a shared vocabulary size of 32k. We train each model on a single A100 GPU and use early stopping with the patience set to 10 epochs on validation loss. We use the development and test sets provided for the English–Icelandic news translation task at WMT 2021 (Akhbardeh et al., 2021), using SacreBLEU.[3] Following Koszowski et al. (2021), we apply regular expressions to fix quotation marks post-translation, making sure Icelandic quotation marks are used in the Icelandic translations and English quotation marks in the English translations.

We train baseline models using the cleaned ParaCrawl dataset and the most recent published version of ParIce, and compare them to models trained on filtered datasets. Table 4 shows resulting BLEU scores. We used the pairwise bootstrap test (Koehn, 2004) to calculate statistical significance. Scores in bold are the highest, but not significantly higher than scores in italics. When models are trained with a cleaner dataset, they seem

[3]SacreBLEU Signature: BLEU+numrefs.1+case.mixed +tok.13a+smooth.exp+version.2.2.0

	ParaCrawl training experiments					ParIce training experiments				
Dataset	no. pairs	en→is	time	is→en	time	no. pairs	en→is	time	is→en	time
Baseline: ParaCrawl v9 clean	2,967,519	20.2	23h3m	30.6	11h14m					
Baseline: ParIce 21.10						1,864,679	19.1	22h54m	25.7	15h06m
Shallow filter 5 - Similar pairs	4,666,464	19.1	18h9m	30.4	29h56m					
Shallow filter 6 - Similar segments	2,081,354	20.0	13h3m	31.9	15h57m					
ParIce shallow filters						1,776,049	*19.7*	23h29m	25.5	14h31m
IS-perplexity (GPT-2)	1,218,256	**21.1**	5h50m	**33.0**	14h11m	1,444,956	18.5	22h33m	24.7	10h18m
SVM	1,991,924	19.6	13h41	32.4	15h56m	1,646,183	*19.8*	17h38m	*26.0*	13h04m
Logistic Regression	1,940,385	20.1	11h48	32.1	12h01m	1,560,346	19.2	16h51m	*26.1*	13h30m
Random Forest	1,981,405	19.5	6h37m	31.8	15h32m	1,667,847	18.6	20h07m	25.2	12h22m
Bicleaner 1.5 (0.50)	1,973,885	19.5	11h25m	32.2	15h33m	1,546,216	*19.5*	21h52m	**26.2**	12h5m
Bicleaner 1.5 (0.67)	1,705,042	19.3	8h29m	31.4	8h53m	1,242,258	*19.5*	12h06m	25.6	9h01m
Bicleaner retrained (0.50)	1,898,209	18.9	8h17m	31.9	15h41m	1,499,610	*19.7*	7h13m	25.6	12h22m
Bicleaner retrained (0.67)	1,615,913	19.5	7h36m	30.5	12h59m	1,244,412	**19.8**	10h16m	25.5	6h13m
Bicleaner lemmatized (0.50)	1,850,884	19.6	10h29m	31.6	17h19m	1,463,780	**19.8**	15h12m	25.9	11h56m
Bicleaner lemmatized (0.67)	1,512,437	19.3	6h27m	30.9	8h32m	1,171,814	**19.8**	7h29m	25.6	8h56m
Bicleaner AI (0.50)	1,235,771	20.5	8h26m	31.7	7h15m	1,262,313	19.1	7h07m	*26.1*	7h44m
Bicleaner AI (0.67)	1,096,288	*21.0*	4h50m	30.8	3h45m	1,152,319	18.9	7h11m	25.1	7h28m

Table 4: BLEU scores and training time for different filtering approaches. Scores in bold are the highest for the dataset and translation direction. Scores in italics are lower, but not significantly lower than the highest ones ($p > 0.05$).

to converge faster, even though the model quality is the same or better. We know from our manual evaluation that most of these training sets contain some erroneous pairs, and in order to try to reduce the number of these, we select the dataset resulting in the highest BLEU score out of the datasets compiled by a Bicleaner model and the best resulting dataset compiled by a classifier. We do an ablation study to investigate whether combining these filters, and the filter looking at perplexity in Icelandic sentences, leads to a better training set. For each dataset and translation direction, we combine the highest scoring Bicleaner model with combinations of the highest scoring statistical classifier and the GPT-2 classifier. Table 5 shows the results for different combinations. For the English→Icelandic translation direction, we obtain higher scores for both corpora using a combination, but for Icelandic→English the BLEU scores never exceed the best standalone filters. We speculate this may be due to noise being more common in the Icelandic texts instigating a need for more filtering when Icelandic is the target language, making the combined filters a better choice in that case, but further investigation is needed.

For our final models, we concatenate the highest-scoring datasets from ParaCrawl and ParIce. These models obtain the highest BLEU

ParaCrawl en→is filters			
Dataset	no. pairs	BLEU	time
Bicleaner AI (0.67) + LogReg	1,071,802	20.4	3h51m
Bicleaner AI (0.67) + GPT-2	776,984	**21.5**	4h17m
Bicleaner AI (0.67) + LogReg + GPT-2	756,503	20.7	3h40m
ParaCrawl is→en filters			
Dataset	no. pairs	BLEU	time
Bicleaner 1.5 (0.50) + SVM	1,930,998	32.3	20h24m
Bicleaner 1.5 (0.50) + GPT-2	1,147,961	31.9	9h02m
Bicleaner 1.5 (0.50) + SVM + GPT-2	1,119,400	32.1	7h32m
ParIce en→is filters			
Dataset	no. pairs	BLEU	time
Bicleaner Lemmatized (0.50) + SVM	1,405,446	**20.2**	17h59m
Bicleaner Lemmatized (0.50) + GPT-2	1,205,070	19.6	14h04m
Bicleaner Lemmatized (0.50) + SVM + GPT-2	1,161,337	18.9	13h24m
ParIce is→en filters			
Dataset	no. pairs	BLEU	time
Bicleaner 1.5 (0.50) + LogReg	1,430,015	*26.1*	13h22m
Bicleaner 1.5 (0.50) + GPT-2	1,269,808	*25.7*	9h30m
Bicleaner 1.5 (0.50) + LogReg + GPT-2	1,179,158	*25.7*	10h46m
Best datasets from both corpora combined			
Dataset	no. pairs	BLEU	time
is→en: ParaCrawl – GPT-2 + ParIce Bicleaner 1.5 (0.50)	2,764,472	***33.2***	15h55m
en→is: ParaCrawl – Bicleaner AI (0.67) + GPT-2 + ParIce – Bicleaner Lemmatized (0.50) + SVM	2,182,430	***22.6***	18h57m

Table 5: BLEU scores and training time for combinations of different filtering approaches. While datasets compiled with combined filters were used to train MT systems delivering the highest BLEU scores for the English→Icelandic translation direction, for Icelandic→English the highest scoring systems were trained on data compiled with only one filter. Scores in bold are the highest scores for the dataset and translation direction they represent. Scores in italics are lower, but not significantly lower ($p > 0.05$). Scores in bold and italics are the highest scores obtained for the translation direction.

scores, 33.2 for Icelandic→English and 22.6 for English→Icelandic. We compare these scores to the results of systems submitted to the WMT 2021 news translation task for the same language pair and directions. Koszowski et al. (2021) submitted a system trained on ParIce and ParaCrawl as well as WikiMatrix and wikititles. The model, based on Transformer$_{\text{BIG}}$ (Vaswani et al., 2017), using back-translation and forward-translation for data augmentation, achieved 22.7 and 33.3 BLEU for en→is and is→en, respectively, only slightly higher, and probably not significantly higher, than our best scores. Símonarson et al. (2021) trained MT models using mBART-25, employing 16 V100 GPUs. They employed back-translations in their training and achieved 22.7 and 32.9 BLEU for en→is and is→en, respectively, after training for 4 days, and after another 4 days and adding more back-translations, they reached 24.3 and 33.5 for en→is and is→en, respectively. These are slightly higher than our best scores. However, we only filter, while they use data augmentation, larger models and more computing power for much longer periods of time.

4 Conclusions and Future Work

In regards to our research questions, our results indicate that different filtering approaches suit different datasets and translation directions, even though we are working within the same language pair. Manual inspection of filtering results and scoring mechanisms seem to be helpful for making informed decisions on how best to filter a dataset. For best results, filtering approaches should be chosen for each translation direction. A limitation of our work is that it does not show which data are detrimental and which are beneficial. In future work, we want to investigate if the differences between datasets used for training can give us an idea of which sentence pairs are most important to filter out. We intend to do this by investigating the pairs discarded by our filters, to compare the data that leads to rising BLEU scores and that which lowers them. This could lead to insights that help constructing filters that work on a more fine-grained level when that is needed.

Our manual evaluation shows that the scores, generated by the automatic scoring systems we employ, have different interpretations depending on the dataset. If scores are used for filtering parallel data, the optimal score should lead to a dataset that produces the best MT model. Feng et al. (2022) suggest a threshold of 0.6 for LaBSE when mining parallel text from CommonCrawl, stating that the threshold was selected by manually inspecting sampled data, but do not specify the language pair used when inspecting the data. In order for the scoring mechanism to be most effective, the user should inspect the results for their dataset before setting a threshold. While all our scoring mechanisms seem to be useful, none of the methods are very good at identifying mutual translations in particular, labelled CC in our taxonomy.

We trained two Bicleaner models for our experiments and our lemmatized model gave the best results for filtering ParIce for the en→is translation direction. The Bicleaner models could perhaps be improved. Bicleaner uses n-gram models and we only used a part of our parallel corpora to train these. If we would use larger corpora the n-gram models would likely give us more accurate scores. The bilingual probability dictionary we used only contained lemmas. By producing all wordforms for the lemmas and trying to estimate the prevalence of each wordform using a monolingual corpus, we could perhaps provide an unlemmatized model with more accurate information leading to better results. Furthermore, we only use 10k sentences to train our GPT-2 perplexity model for Icelandic. A larger dataset may increase its accuracy.

Two systems participating in the WMT 2021 news translation task, evaluated on the same data, obtain scores only slightly higher than ours, but while we only train a Transformer$_{\text{BASE}}$ model, they train larger models using more resources and much longer training time. In our experiments, models that have been filtered more tend to converge faster. We can deduce from this that training data that is better filtered, not only improves MT output quality, but is also in line with a call to greener and more sustainable models of AI, see e.g. Yusuf et al. (2021) and Jooste et al. (2022).

Acknowledgements

This work was supported by the The Icelandic Centre for Research, RANNIS grant number 228654-051, and by the ADAPT Centre for Digital Content Technology which is funded under the Science Foundation Ireland (SFI) Research Centres Programme (Grant No. 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 Conference on Machine Translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online.

Lora Aroyo, Matthew Lease, Praveen Paritosh, and Mike Schaekermann. 2022. Data Excellence for AI: Why Should You Care? *Interactions*, 29(2):66–69.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas.

Mikel Artetxe and Holger Schwenk. 2019a. Margin-based Parallel Corpus Mining with Multilingual Sentence Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy.

Mikel Artetxe and Holger Schwenk. 2019b. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A Configurable Parallel Corpus Filtering Toolbox. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 150–156, Online.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online.

Starkaður Barkarson and Steinþór Steingrímsson. 2019. Compiling and Filtering ParIce: An English-Icelandic Parallel Corpus. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 140–145, Turku, Finland.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 Conference on Machine Translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online.

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45(1):5–32.

Eleftheria Briakou and Marine Carpuat. 2021. Beyond Noise: Mitigating the Impact of Fine-grained Semantic Divergences on Neural Machine Translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7236–7249, Online.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-Resource Corpus Filtering Using Multilingual Sentence Embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 261–266, Florence, Italy.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

David R Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232.

Arne Defauw, Sara Szoc, Anna Bardadym, Joris Brabers, Frederic Everaert, Roko Mijic, Kim Scholte, Tom Vanallemeersch, Koen Van Winckel, and Joachim Van den Bogaert. 2019. Misalignment Detection for Web-Scraped Corpora: A Supervised Regression Approach. *Informatics*, 6(3):35.

Georgiana Dinu and Marco Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek,

Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond English-Centric Multilingual Machine Translation. *J. Mach. Learn. Res.*, 22(1).

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT Sentence Embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland.

Christian Herold, Jan Rosendahl, Joris Vanvinckenroye, and Hermann Ney. 2022. Detecting Various Types of Noise for Neural Machine Translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2542–2551, Dublin, Ireland.

Wandri Jooste, Rejwanul Haque, and Andy Way. 2022. Knowledge Distillation: A Method for Making Neural Machine Translation More Efficient. *Information*, 13(2).

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain.

Huda Khayrallah and Philipp Koehn. 2018. On the Impact of Various Types of Noise on Neural Machine Translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the WMT 2020 Shared Task on Parallel Corpus Filtering and Alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742, Online.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 Shared Task on Parallel Corpus Filtering for Low-Resource Conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 Shared Task on Parallel Corpus Filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels.

Mikołaj Koszowski, Karol Grzegorczyk, and Tsimur Hadeliya. 2021. Allegro.eu Submission to WMT21 News Translation Task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 140–143, Online.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual Word Representations with Monolingual Quality in Mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, Denver, Colorado.

Mark Mazumder, Colby R. Banbury, Xiaozhe Yao, Bojan Karlavs, William Gaviria Rojas, Sudnya Diamos, Gregory Frederick Diamos, Lynn He, Douwe Kiela, David Jurado, David Kanter, Rafael Mosquera, Juan Ciro, Lora Aroyo, Bilge Acun, Sabri Eyuboglu, Amirata Ghorbani, Emmett D. Goodman, Tariq Kane, Christine R. Kirkpatrick, Tzu-Sheng Kuo, Jonas Mueller, Tristan Thrush, Joaquin Vanschoren, Margaret J. Warren, Adina Williams, Serena Yeung, Newsha Ardalani, Praveen K. Paritosh, Ce Zhang, James Y. Zou, Carole-Jean Wu, Cody Coleman, Andrew Y. Ng, Peter Mattson, and Vijay Janapa Reddi. 2022. DataPerf: Benchmarks for Data-Centric AI Development. *ArXiv*, abs/2207.10062.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Mārcis Pinnis. 2018. Tilde's Parallel Corpus Filtering Methods for WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 939–945, Belgium, Brussels.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz Rojas. 2020. Bifixer and Bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal.

Spencer Rarrick, Chris Quirk, and Will Lewis. 2011. MT Detection in Web-Scraped Parallel Corpora. In *Proceedings of Machine Translation Summit XIII: Papers*, pages 422–429, Xiamen, China.

Matiss Rikters. 2018. Impact of Corpora Quality on Neural Machine Translation. In *Human Language Technologies–The Baltic Perspective: Proceedings of the Eighth International Conference Baltic HLT 2018*, pages 126–133, Tartu, Estonia. IOS Press.

Nick Rossenbach, Jan Rosendahl, Yunsu Kim, Miguel Graça, Aman Gokrani, and Hermann Ney. 2018. The RWTH Aachen University Filtering System for the WMT 2018 Parallel Corpus Filtering Task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 946–954, Belgium, Brussels.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez. 2018. Prompsit's submission to WMT 2018 Parallel Corpus Filtering shared task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 955–962, Belgium, Brussels.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Haukur Barri Símonarson, Vésteinn Snæbjarnarson, Pétur Orri Ragnarson, Haukur Jónsson, and Vilhjalmur Thorsteinsson. 2021. Miðeind's WMT 2021 Submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 136–139, Online.

Steinþór Steingrímsson and Starkaður Barkarson. 2021. http://hdl.handle.net/20.500.12537/145 ParIce: English-Icelandic parallel corpus (21.10). CLARIN-IS.

Steinþór Steingrímsson, Sigrún Helgadóttir, Eiríkur Rögnvaldsson, Starkaður Barkarson, and Jón Guðnason. 2018. Risamálheild: A Very Large Icelandic Text Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, LREC 2018, pages 4361–4366, Miyazaki, Japan.

Steinþór Steingrímsson, Pintu Lohar, Hrafn Loftsson, and Andy Way. 2021. Effective Bitext Extraction From Comparable Corpora Using a Combination of Three Different Approaches. In *Proceedings of the 14th Workshop on Building and Using Comparable Corpora (BUCC 2021)*, pages 8–17, Online (Virtual Mode).

Steinþór Steingrímsson, Luke O'Brien, Finnur Ingimundarson, Hrafn Loftsson, and Andy Way. 2022. Compiling a Highly Accurate Bilingual Lexicon by Combining Different Approaches. In *Proceedings of Globalex Workshop on Linked Lexicography within the 13th Language Resources and Evaluation Conference*, pages 32–41, Marseille, France.

Jannis Vamvas and Rico Sennrich. 2022. NMTScore: A multilingual analysis of translation-based text similarity measures. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 198–213, Abu Dhabi, United Arab Emirates.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5999–6009, Long Beach, California.

Yinfei Yang, Gustavo Hernandez Abrego, Steve Yuan, Mandy Guo, Qinlan Shen, Daniel Cer, Yun-hsuan

Sung, Brian Strope, and Ray Kurzweil. 2019. Improving Multilingual Sentence Embedding using Bi-directional Dual Encoder with Additive Margin Softmax. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5370–5378, Macao, China.

Mirza Yusuf, Praatibh Surana, Gauri Gupta, and Krithika Ramesh. 2021. Curb Your Carbon Emissions: Benchmarking Carbon Emissions in Machine Translation. *ArXiv*, abs/2109.12584.

Jaume Zaragoza-Bernabeu, Gema Ramírez-Sánchez, Marta Bañón, and Sergio Ortiz Rojas. 2022. Bicleaner AI: Bicleaner Goes Neural. In *Proceedings of the Language Resources and Evaluation Conference*, pages 824–831, Marseille, France.

Gamli – Icelandic Oral History Corpus: Design, Collection and Evaluation

Luke O'Brien
Tiro
luke@tiro.is

Finnur Ágúst Ingimundarson
The Árni Magnússon Institute
for Icelandic Studies
fai@hi.is

Jón Guðnason
University
of Reykjavik
jg@ru.is

Steinþór Steingrímsson
The Árni Magnússon Institute
for Icelandic Studies
steinthor.steingrimsson
@arnastofnun.is

Abstract

We present *Gamli*, an ASR corpus for Icelandic oral histories, the first of its kind for this language, derived from the Ísmús ethnographic collection. Corpora for oral histories differ in various ways from corpora for general ASR, they contain spontaneous speech, multiple speakers per channel, noisy environments, the effects of historic recording equipment, and typically a large proportion of elderly speakers. Gamli contains 146 hours of aligned speech and transcripts, split into a training set and a test set. We describe our approach for creating the transcripts, through both OCR of previous transcripts and post-editing of ASR output. We also describe our approach for aligning, segmenting, and filtering the corpus and finally training a Kaldi ASR system, which achieves 22.4% word error rate (WER) on the Gamli test set, a substantial improvement from 58.4% word error rate from a baseline general ASR system for Icelandic.

1 Introduction

Icelandic open-licensed speech corpora have in recent years grown in volume and numbers, there are now *Talrómur* (Sigurgeirsson et al., 2021), *Málrómur* (Steingrímsson et al., 2017), *Samrómur* (Mollberg et al., 2020) and the Althingi's Parliamentary Speeches corpus (Helgadóttir et al., 2017; Nikulásdóttir et al., 2018) to name a few. However both historical speech and older speakers are underrepresented in these corpora. For instance, regarding older speakers, in Samrómur, the largest open-licensed ASR corpus for Icelandic (2233 hours in the latest release, Hedström et al. 2022), only 4.8% of speakers are over 60 years old.

Gamli, the oral history speech corpus presented in this paper differs from that in many ways. Firstly, it contains, predominantly, spontaneous speech in the form of interviews, secondly, it has a very high ratio of older speakers (94.8% of speakers are over 60 years old), thirdly, background noise is common as well as noise artefacts from historical recording equipment and lastly, historic dialects (word choice and accent) are much more prevalent than in existing corpora.

The corpus contains 146 hours of aligned speech and transcripts split into a training set and a test set. This data, based on valuable historical 20th century recordings stored at the Department of Ethnology and Folklore at The Árni Magnússon Institute for Icelandic Studies, is therefore an important addition to the existing Icelandic speech corpora.[1]

The custom ASR system presented in this paper along with the corpus will in due course be used to automatically transcribe all of the ethnographic audio recordings stored at the institute. The transcripts will then be made available on the online portal *Ísmús*[2] and paired with the respective recording.

2 Related Work

For many years, ASR systems have been trained on unaligned transcriptions (Panayotov et al., 2015) and even approximate transcriptions of spontaneous speech (Jang and Hauptmann, 1999). In the case of Icelandic ASR for spontaneous speech, there has been an ongoing project (Helgadóttir et al., 2017), (Helgadóttir et al., 2017) to align and filter Icelandic parliamentary transcripts for ASR in order to reduce the manual work involved in transcribing parliamentary proceedings.

[1] The corpus is available under an open license at http://hdl.handle.net/20.500.12537/310

[2] www.ismus.is

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 601–609

Creating the corpora involves text normalization, time-alignment, and filtering utterances.

While ASR for oral histories is new for Icelandic, it is already being used in other languages. For example, the first large project was the MALACH project (Psutka et al., 2002) in 2002, where ASR transcriptions were used for indexing oral history archives and making them more searchable. However, some authors still consider oral history speech recognition an open problem (Picheny et al., 2019; Gref et al., 2020) and a recent study (Gref et al., 2022) found that human word error rate was 8.7% on a German oral history corpus (taking into account case-sensitivity and annotation of hesitations). Whereas Lippmann (1997) found a human word error rate of less than 4% on the Switchboard corpus of spontaneous telephony speech and less than 0.4% on the Wall Street Journal corpus of clear read speech. This suggests that the minimum possible word error rate for ASR might be much higher on oral histories than it is for cleaner speech corpora.

One other factor that makes oral history ASR an interesting challenge is the particularly high ratio of older speakers. It has been noted by Vipperla et al. (2008) that for general ASR models, WER correlates strongly with age, even throughout a single speakers lifetime. This could be caused by multiple changes in aging voices, such as slower speaking rate, changes in F0 (decrease for males and increase for females), increase in jitter and shimmer (all from Vipperla et al. (2008)), some of which could be mitigated by increasing the number of older speakers in the training set. However, other changes might not be so easily solved, such as a reduction of tongue and jaw strength and an increase in breathiness (all from Vipperla et al. (2008)) which could reduce articulatory precision.

There are three main use-cases for oral history speech recognition. First, to index oral archives for spoken document retrieval. Second, to provide transcripts to aid listeners. Third, as a hypothesis transcript for post-editing. For each of these use-cases, it's important to determine the minimum acceptable ASR performance. For the first use-case, indexing, Chelba et al. (2008) found that using ASR output significantly improves spoken document retrieval performance compared to only using the accompanying text meta data, even when WER is as high as 50%. More recently, Fan-Jiang et al. (2020) used a BERT-based retrieval model with query reformulation and managed to get impressive results for document retrieval of Mandarin news when using erroneous recognition transcripts (35%). The accuracy was 0.594 with the erroneous transcripts and 0.597 with reference transcripts. This suggests that for indexing, acceptable WER may be even higher than 35%. For the second use-case, to provide transcriptions as an aid to listeners, Munteanu et al. (2006) found that transcripts with a 25% WER improved listeners' understanding more than listening to audio without a transcript, however they found that understanding was reduced when the transcripts had 45% WER, suggesting that a maximum acceptable WER is somewhere between 25% and 45%. For the third use-case, post-editing, Gaur et al. (2016) found that for recordings of Ted Talks, ASR transcriptions with less than 30% WER sped up the transcription process but if the WER was higher than 30% it slowed transcribers down.

3 Origin of the corpus

The ethnography collection of the Department of Ethnology and Folklore at The Árni Magnússon Institute for Icelandic Studies contains more than 2,300 hours of audio recordings of oral heritage and traditions, with a little less than 2,500 interviewees. The oldest material are recordings made on wax cylinders in the early 20th century and the collection is continually expanding with new material being added every year.

The bulk of the collection, however, consists of recordings from the 1960's and 1970's, mainly the work of three collectors. Their focus was to gather ethnographic material from all of Iceland, first and foremost from older generations — the majority of the informants were born before or around the turn of the 20th century,

This resulted in an extensive collection of legends and fairy tales, accounts of beliefs and customs, poems, hymns, nursery rhymes, Icelandic ballads (*rímur*), occasional verses and more, with the material being variously spoken, sung or chanted. Apart from recited verse and that which is sung or chanted the speech is spontaneous. Accompanying the recordings is detailed metadata on the speaker, time and location of recording, as well as various other parameters such as genre (for different kinds of verse or prose material, e.g. poems or nursery rhymes, fairy tales or legends etc.), mode of performance (sung, chanted, spoken), key

words, content (short summary, description), tale-types and motifs (in folktales and legends).

3.1 Speaker distribution in the collection

In their work the collectors mainly relied on a snowball method of sorts, asking speakers to point them to other possible informants, as well as contacting teachers or clergy to enquire about interesting subjects in their region. Speaker profession is often listed in the metadata and most of the speakers were workers, farmers, fishermen, housewives etc., with little formal education.

Gender was probably not a decisive factor at the outset and the total ratio is 57.6% male speakers and 42.4% female, i.e. based on the number of speakers. However, if audio length for each gender is included the difference increases quite a bit, i.e. 1504 hours (65%) for men vs. 821 hours (35%) for women.

As mentioned, the data in the collection also stands out in that that the age of the speakers is higher than in other existing Icelandic corpora. The oldest speaker in the collection was 105 years old at the time of recording in 1954 and the oldest speaker in the collection, with regards to date of birth, was born in 1827, and recorded in 1904 (not included in the *Gamli* corpus). In fact, 72.4% of the speakers are older than 63 and 31.4% are 71-80 years old. In *Gamli* this ratio is substantially higher, as detailed in Section 4.

3.2 Regional features in pronunciation

The speakers in the collection are from all over Iceland and therefore reflect the various regional differences in pronunciation much better than recently recorded speech corpora such as *Samrómur*, due to the fact that these regional features either have already more or less disappeared or are gradually disappearing. Amongst these features is for example the "hard" pronunciation of /p, t, k/ (still a distinct feature) and voiced pronunciation of /l, m, n/ before /p, t, k/ in North-Iceland, *rn*-, *rl*-pronunciation in South-East-Iceland, monophthongs before /ng, nk/ in the North-West etc.

While these features are not tagged in any way in the *Gamli* corpus, the ASR system trained on the corpus seems to work well on these features, with possibly the exception of labial or velar stops before [ð], such as [hapðɪ] instead of [havðɪ] for *hafði* or [lakðɪ] instead of [laɣðɪ] for *lagði*. We have, however, not inspected this systematically, so it needs further looking into to state the precision with any certainty.

3.3 Recording procedure

Most of the recordings were made at the speakers' homes, in many cases in elderly homes, and carried out by the interviewer. It was not uncommon that other people, e.g. children, spouses etc. were present during the recording sessions, but they were in most cases not meant to play a part in the recording. Because of this, and for various other reasons, some background noise and disturbances occur in the recordings, e.g. children playing, traffic sounds, phones ringing etc., but these are generally not prominent.

Much of the recordings were recorded using high quality reel-to-reel tape recording devices, although some were done by amateurs who weren't as well equipped, whereas a part of the recordings are from the recording studios of The Icelandic National Broadcasting Service (Þorsteinsdóttir, 2013).

The digitalization of these recordings began in the late 1990's and continued into the early 2000's with the recordings being converted into WAV format as well as compressed MP3s for online use.

4 Corpus content

Gamli contains 146 hours of transcribed audio broken down into

1. $\sim$ 111 hours from optical character recognition (OCR) of previous transcriptions in various formats
2. $\sim$ 35 hours of new transcriptions (post-edited from ASR output)

The 111 hours include 9 hours defined as a test set, which was manually reviewed and corrected and annotated with speaker ID and time alignments in the annotation tool *ELAN*. The test set contains recordings with 10 speakers, 5 women and 5 men, plus the interviewers (4 men) and serves for evaluating the system's performance.

A validation set has not been defined for the corpus as the acoustic model training in *Kaldi* (Povey et al., 2011) used a random sample of the training corpus for validation.

4.1 Speaker distribution in the corpus

The corpus contains 210 unique speakers, 90 women and 120 men (plus the interviewers: 14

Data split	Hours	Male speakers	Female speakers	Total speakers
Training	137	115	85	200
Test	9	5	5	10

Table 1: Data splits in Gamli

men and 1 woman). At the outset we aimed to have the gender ratio as equal as possible in the acoustic training data, but with three men surpassing 20 hours of speech each (with one topping at 29 hours) and accounting for more than one third of the entire data, that picture became quite distorted. As a result the gender bias in the corpus is even greater than in the collection itself, which is unfortunate, but simply reflects the data that was at hand, i.e. 73.5% vs. 26.5%, cf. Section 4.2.

The age ranges from 38 to 99, but most of the speakers are 60+ (94.8%), as shown in Figure 1, and the average age of the speakers is 77 years. This ratio is unprecedented in all existing corpora for Icelandic speech (cf. 4.8% in Samrómur as referred to in Section 1) and makes Gamli an important addition to that collection.

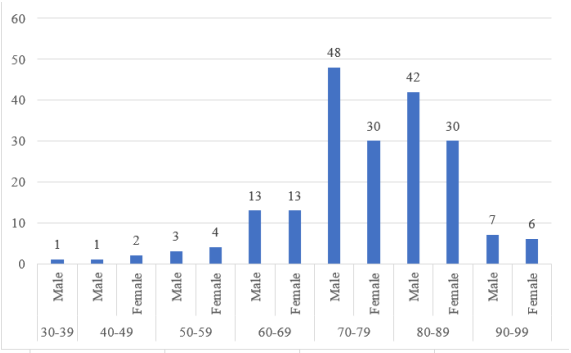


Figure 1: Age distribution of unique speakers in the training set

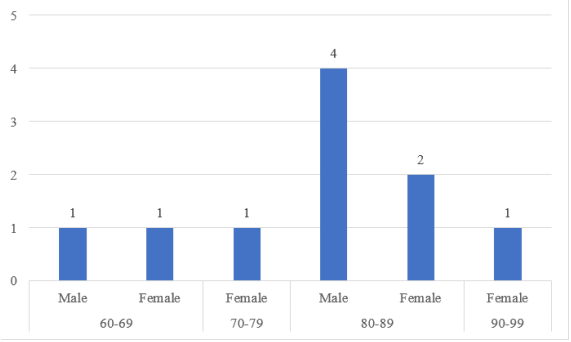


Figure 2: Age distribution of unique speakers in the test set

4.2 Corpus compilation

As mentioned, the largest part of the corpus, about 111 hours, stems from OCR of transcriptions at the Department of Ethnology and Folklore at The Árni Magnússon Institute for Icelandic Studies. These transcripts that were generated over several decades are not all in the same format (e.g. typewritten, dot printed, printed Word documents) and therefore needed first to be processed, i.e. scanned and OCRed (the results of which varied depending on the format). These transcripts were then catalogued and paired with the respective recordings.

Once this ready data had been processed the first ASR output was produced and manually corrected. During that process it became evident that some of the recordings were ill suited at this stage as they often contained poetry, nursery rhymes and in some cases singing, where the ASR system could not be expected to do well as the focus was on spontaneous speech, where it performed much better (cf. Section 6).

As a result, we made use of the detailed metadata search parameters in the Ísmús portal in order to filter the best in-domain data for further training. We mainly relied on the so-called *form* parameter (*genre*) to try to exclude everything but spontaneous speech. This gave much better results and resulted in the 35 hours of post-edited data mentioned in Section 4.

4.3 Normalizing, aligning, segmenting and filtering the transcripts for ASR training

The transcripts in the training set did not have time alignments and some had OCR spelling errors. Therefore, we had to process the transcripts before using them to train the acoustic model. First, the text was normalized using the Regina normalizer developed in Sigurðardóttir (2021). Second, the text was aligned to the audio with Kaldi's segment long utterances function [3]. For this, a biased language model (based on the text) is combined with an existing acoustic model to force-align the audio, as detailed in section 2.2 of Manohar et al. (2017). It outputs aligned segments of less than 15 seconds each. Third, these segments are filtered with Kaldi's clean and segment data function [4] which again combines a biased language model

[3] https://github.com/kaldi-asr/kaldi/blob/master/egs/wsj/s5/steps/cleanup/segment_long_utterances_nnet3.sh

[4] https://github.com/kaldi-asr/kaldi/blob/master/egs/wsj/s5/steps/cleanup/

(based on the text) with an existing acoustic model and removes segments that were unintelligible to the decoder.

After filtering, 180 hours of interviews was reduced to 137 hours (24% reduction). However, much of this reduction can be attributed to silences in the audio, so to estimate the total amount of speech reduced, we note that the word count was reduced from 1,147,181 to 1,039422 (9.4% reduction).

Finally, after training an acoustic model on this in-domain data, the alignment, segmentation, and filtering was performed again. That final data constitutes the Gamli training set. The final model was then trained on that data.

5 Models (and out-of-domain data)

We trained a hybrid ASR system in Kaldi. That is, the language model and acoustic model were trained separately as opposed to an end-to-end system. For the acoustic and language models in the custom ASR system, we expanded the training sets with various out-of-domain data, which will be described in the following sections.

5.1 Acoustic Model

An acoustic model learns to map audio to a sequence of phonemes. The acoustic model is a TDNN (time-delayed neural network) chain model trained in Kaldi. It was trained on the in-domain data described above, but also on various out-of-domain data, which included the following datasets:

1. Althingi's Parliamentary Speeches.[5] A corpus of 514.5 hours of recorded speech from the Icelandic parliament (Helgadóttir et al., 2017)

2. 114.6 hours of speech from the first Samrómur release,[6] leaving out children.

3. 173.1 hours of unverified Samrómur data,[7] containing only speech with 50+ year old men and 60+ year old women.

4. 228.2 hours of the RÚV TV unknown speakers dataset.[8]

iVectors and MFCCs (Mel-frequency cepstral coefficients) are the inputs to the acoustic model. These are commonly used in Kaldi 'chain' models. The iVectors in particular are said to make the neural network speaker adaptive since the vectors themselves carry speaker identity information (Saon et al., 2013).

Data augmentation was also used to triple the entire training set. We added artificial noise and reverberation. For noisy data sets, e.g. call-center data sets, this is said to give better results than speed perturbations (Ko et al., 2017) and as was described earlier, background noise and disturbances are not uncommon in the data.

5.2 Language Model

A language model is necessary for outputting coherent texts, it learns a probability distribution for word sequences from a training corpus. The language modelling consists of a 3-gram language model for decoding and an RNN language model for rescoring. It was trained on the Gamli training set described in 4.2, as well as out-of-domain data. The out-of-domain data stems from the following sources:

1. The Icelandic Gigaword Corpus (IGC) (Steingrímsson et al., 2018). We use the sentences from the 2022 version of the IGC.[9]

2. Ethnographic data from the National Museum of Iceland in *Sarpur*.[10]

3. Audio file descriptions from Ísmús [11] for their content.

4. Place name data from the Icelandic Place Name Collection.[12]

5.3 Vocabulary and Pronunciation Dictionary

The pronunciation dictionary maps words to sequences of phonemes. For the vocabulary we used:

clean_and_segment_data_nnet3.sh

[5] Available at: http://hdl.handle.net/20.500.12537/277

[6] Available at: http://hdl.handle.net/20.500.12537/189

[7] Available at: http://hdl.handle.net/20.500.12537/265

[8] Available at: http://hdl.handle.net/20.500.12537/191

[9] http://hdl.handle.net/20.500.12537/254

[10] https://sarpur.is/

[11] https://ismus.is/

[12] nafnid.is

1. All the word forms from *The Database of Icelandic Morphology* (Bjarnadóttir et al., 2019).

2. OOV words from audio file descriptions in Ísmús.

3. Vocabulary from the training set (only the data that was manually transcribed and not the OCR data); manually checked and added where appropriate.

4. OOV words from *Sarpur*; (manually checked and added where appropriate).

To get the phonemic transcriptions of each word a G2P model based on the *Icelandic Pronunciation Dictionary for Language Technology*[13] was used.

6 Evaluation

To assess the final ASR system's performance on the test set, we compare it to two baselines. The first is the out-of-domain system, which was trained in the same way as the final system but only on the out-of-domain data detailed in sections 5.1 and 5.2, not on the Gamli training set. The second baseline is the Samrómur "base" system [14]. This is a kaldi-trained system from a well-known dataset of read Icelandic speech, the acoustic mode is a TDNN chain model the language model is an n-gram model. While the ASR baseline systems achieved 36.7% and 58.4% respectively on the Gamli test set, the final ASR system performed better, achieving 22.4% WER on the same set, as shown in Table 2. This table compares the three overall systems, each including their own acoustic model and language model. However, it should be noted that the same lexicon and vocabulary were used for the final system and the out-of-domain system.

To investigate the differences in the systems, we also compare the performance when taking demographic information into account in Figure 3. As stated earlier, the test set contains 10 speakers and a total of 9 hours of audio.

To separate the effect that the Gamli training set had on acoustic model adaptation and language model adaptation, in Table 3, we compare WER when combining the out-of-domain models with the final models, using the same lexicon and vocabulary.

[13] Available at: http://hdl.handle.net/20.500.12537/99

[14] https://github.com/cadia-lvl/samromur-asr/tree/master/s5_base

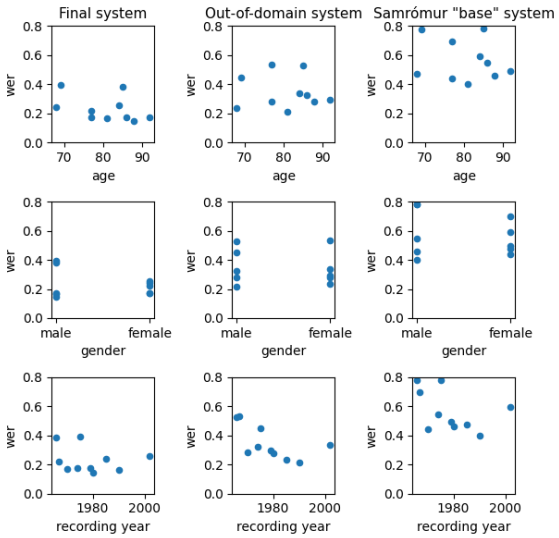


Figure 3: WER for the 10 unique speakers in the Gamli test set based on demographic information. Comparing the final system we trained, the out-of-domain system we trained, and the Kaldi-based Samrómur "base" system

	WER	OOV-rate total words	OOV-rate unique words
Baseline (Samrómur)	58.4%	1.1%	6.8%
Out-of-domain	36.7%	0.5%	3.1%
Final	**22.4%**	0.5%	3.1%

Table 2: ASR performance on the Gamli oral history test set

	Out-of-domain LM	Final LM
Out-of-domain AM	36.7%	34.1%
Final AM	24.0%	**22.4%**

Table 3: ASR performance (WER) on the Gamli oral history test set when combining a specific acoustic model with a specific language model. Note that the final models were trained on the Gamli training set, while the out-of-domain models were not

It seems that acoustic model adaptation had a larger impact than language model adaptation for WER on the Gamli test set.

This is an interesting finding, seeing as language model adaptation is generally more commonly performed, at least in Kaldi, where it takes less computing power than acoustic model adaptation. Though, the results from Table 3 could sim-

ply be due to particularly good out-of-domain text data, they could also suggest that the acoustic elements of oral history are particularly different to other ASR datasets available for Icelandic, and if this is the case, the Gamli training set could be a useful addition to the currently available Icelandic data in order to make acoustic models more robust to elderly speech, historic speech, and historic recording equipment.

7 Conclusion and Future Work

In this paper we have presented *Gamli*, a corpus suitable for training speech recognition systems, we have aligned and segmented Icelandic oral histories from manual transcriptions (both OCR from typewritten transcripts and post-edited from ASR output), and filtered out unintelligible segments.

We have described the compilation of the corpus, which has been published under an open license, the origins of the data and evaluation of an ASR system trained on the corpus. We have shown that using the corpus along with other relevant datasets can substantially lower WER for historical speech data, from 58.4% from a baseline system (Samrómur "base" system) to 22.4%. We also draw the conclusion that it could be combined with other ASR training sets which lack in historical recordings and speech from older speakers in order to improve robustness to such audio.

Our final ASR system will be used to automatically transcribe the entire ethnographic audio data stored in Ísmús, i.e. 2,300 hours of audio. We expect the outcome of that process to be in line with the results presented in this paper, with verse, nursery rhymes, singing etc. still remaining a challenge for the customised model, but accuracy for spontaneous speech to be more reliant on audio quality and clarity of speech. Where the quality of these two factors is high, we expect the system to perform well.

Even though the WER may differ substantially for some files, the general outcome will nonetheless be a somewhat readable version of the Ísmús ethnographic collection. As outlined in 1, that output can subsequently be used in a number of ways: first, indexing the Ísmús ethnographic collection for search queries (useful for longer audio files where the description can not do the entire content justice). Second, presenting transcripts alongside the audio as a listening aid and to increase accessibility. Third, as a hypothesis transcript for post-editing of more transcripts.

The *Gamli* corpus itself should provide an interesting challenge to linguists and ASR researchers interested in spontaneous speech, older speakers, noisy audio, historical recordings and historical dialects.

8 Acknowledgements

This paper and the project it is based on are funded by a grant from the Infrastructure Fund from the The Icelandic Centre for Research in collaboration with the Center for Digital Humanities (*Miðstöð stafrænna hugvísinda og lista*) at the University of Iceland. We would like to thank the following people for their collaboration and contribution to the project: Eydís Huld Magnúsdóttir, CEO of Tiro, Rósa Þorsteinsdóttir, curator of the ethnographic collection at the Árni Magnússon Institute for Icelandic Studies and main editor of Ísmús, and Trausti Dagsson, the manager of this project and programmer at the same institute. We would also like to thank our three anonymous reviewers for their comments. Lastly, we thank all the interviewers and interviewees, even though most of them have now gone on to the great beyond, for their invaluable contribution.

References

Kristín Bjarnadóttir, Kristín Ingibjörg Hlynsdóttir, and Steinþór Steingrímsson. 2019. DIM: The Database of Icelandic Morphology. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 146–154, Turku, Finland.

Ciprian Chelba, Timothy J. Hazen, and Murat Saraclar. 2008. Retrieval and Browsing of Spoken Content. *IEEE Signal Processing Magazine*, 25(3):39–49.

Shao-Wei Fan-Jiang, Tien-Hong Lo, and Berlin Chen. 2020. Spoken Document Retrieval Leveraging Bert-Based Modeling and Query Reformulation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8144–8148.

Yashesh Gaur, Walter S Lasecki, Florian Metze, and Jeffrey P Bigham. 2016. The effects of automatic speech recognition quality on human transcription latency. In *Proceedings of the 13th International Web for All Conference*, pages 1–8.

Michael Gref, Nike Matthiesen, Sreenivasa Hikkal Venugopala, Shalaka Satheesh, Aswinkumar Vijayananth, Duc Bach Ha, Sven Behnke, and Joachim Köhler. 2022. A Study on the Ambiguity in Human Annotation of German Oral History

Interviews for Perceived Emotion Recognition and Sentiment Analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2022–2031, Marseille, France.

Michael Gref, Oliver Walter, Christoph Schmidt, Sven Behnke, and Joachim Köhler. 2020. Multi-Staged Cross-Lingual Acoustic Model Adaption for Robust Speech Recognition in Real-World Applications - A Case Study on German Oral History Interviews. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6354–6362, Marseille, France.

Staffan Hedström, Judy Y. Fong, Ragnheiður Þórhallsdóttir, David Erik Mollberg, Smári Freyr Guðmundsson, Ólafur Helgi Jónsson, Sunneva Þorsteinsdóttir, Eydís Huld Magnúsdóttir, and Jon Gudnason. 2022. Samromur Unverified 22.07. CLARIN-IS.

Inga Rún Helgadóttir, Róbert Kjaran, Anna Björk Nikulásdóttir, and Jón Guðnason. 2017. Althingi's Parliamentary Speeches. CLARIN-IS.

Inga Rún Helgadóttir, Róbert Kjaran, Anna Björk Nikulásdóttir, and Jón Guðnason. 2017. Building an ASR Corpus Using Althingi's Parliamentary Speeches. In *Proceedings of Interspeech 2017*, pages 2163–2167, Stockholm, Sweden.

Photina Jaeyun Jang and Alexander G. Hauptmann. 1999. Improving Acoustic Models with Captioned Multimedia Speech. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, volume 2 of *ICMCS '99*, pages 767–771, USA.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur. 2017. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5220–5224. IEEE.

Richard P. Lippmann. 1997. Speech recognition by machines and humans. *Speech Communication*, 22(1):1–15.

Vimal Manohar, Daniel Povey, and Sanjeev Khudanpur. 2017. Jhu kaldi system for arabic mgb-3 asr challenge using diarization, audio-transcript alignment and transfer learning. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 346–352. IEEE.

David Erik Mollberg, Ólafur Helgi Jónsson, Sunneva Þorsteinsdóttir, Steinþór Steingrímsson, Eydís Huld Magnúsdóttir, and Jon Gudnason. 2020. Samrómur: Crowd-sourcing Data Collection for Icelandic Speech Recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3463–3467, Marseille, France.

Cosmin Munteanu, Ronald Baecker, Gerald Penn, Elaine Toms, and David James. 2006. The Effect of Speech Recognition Accuracy Rates on the Usefulness and Usability of Webcast Archives. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, page 493–502, Montréal, Québec.

Anna Björk Nikulásdóttir, Inga Rún Helgadóttir, Matthías Pétursson, and Jón Guðnason. 2018. Open ASR for Icelandic: Resources and a Baseline System. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3137–3141, Miyazaki, Japan.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, South Brisbane, QLD, Australia.

Michael Picheny, Zoltán Tüske, Brian Kingsbury, Kartik Audhkhasi, Xiaodong Cui, and George Saon. 2019. Challenging the Boundaries of Speech Recognition: The MALACH Corpus. In *Proceedings of Interspeech 2019*, Graz, Austria.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Josef Psutka, Pavel Ircing, Josef V Psutka, Vlasta Radová, William J Byrne, Jan Hajič, Samuel Gustman, and Bhuvana Ramabhadran. 2002. Automatic transcription of Czech language oral history in the MALACH project: Resources and initial experiments. In *Text, Speech and Dialogue: 5th International Conference, TSD 2002 Brno, Czech Republic, September 9–12, 2002 Proceedings 5*, pages 253–260. Springer.

George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. 2013. Speaker Adaptation of Neural Network Acoustic Models Using I-Vectors. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 55–59, Olomouc, Czech Republic.

Helga Svala Sigurðardóttir. 2021. Text normalization corpus 21.10 (2021-10-25). CLARIN-IS.

Atli Sigurgeirsson, Þorsteinn Gunnarsson, Gunnar Örnólfsson, Eydís Magnúsdóttir, Ragnheiður Þórhallsdóttir, Stefán Jónsson, and Jón Guðnason. 2021. Talrómur: A large Icelandic TTS corpus. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 440–444, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Steinþór Steingrímsson, Jón Guðnason, Sigrún Helgadóttir, and Eiríkur Rögnvaldsson. 2017. Málrómur: A Manually Verified Corpus of Recorded Icelandic Speech. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 237–240, Gothenburg, Sweden.

Steinþór Steingrímsson, Sigrún Helgadóttir, Eiríkur Rögnvaldsson, Starkaður Barkarson, and Jón Guðnason. 2018. Risamálheild: A Very Large Icelandic Text Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, LREC 2018, pages 4361–4366, Miyazaki, Japan.

Ravichander Vipperla, Steve Renals, and Joe Frankel. 2008. Longitudinal study of ASR performance on ageing voices. In *Proceedings of Interspeech 2008*, pages 2550–2553, Brisbane, Australia.

Rósa Þorsteinsdóttir. 2013. Ísmús (Íslenskur músík- og menningararfur): An Open-Access Database. *The Retrospective Methods Network Newsletter*, 7:97–101.

NoCoLA: The Norwegian Corpus of Linguistic Acceptability

Matias Jentoft and **David Samuel**
University of Oslo, Language Technology Group
{matiasj, davisamu}@ifi.uio.no

Abstract

While there has been a surge of large language models for Norwegian in recent years, we lack any tool to evaluate their understanding of grammaticality. We present two new Norwegian datasets for this task. **NoCoLA**$_{class}$ is a supervised binary classification task where the goal is to discriminate between acceptable and non-acceptable sentences. On the other hand, **NoCoLA**$_{zero}$ is a purely diagnostic task for evaluating the grammatical judgement of a language model in a completely zero-shot manner, i.e. without any further training. In this paper, we describe both datasets in detail, show how to use them for different flavors of language models, and conduct a comparative study of the existing Norwegian language models.

1 Introduction

Large pre-trained language models have recently led to a revolution in natural language processing (NLP) as they substantially increased the performance of most NLP tools (Peters et al., 2018; Devlin et al., 2019). Large language models were originally developed for English, but a surge of Norwegian-based models has recently followed (Kutuzov et al., 2021; Kummervold et al., 2021; Hofmann et al., 2022). The remaining issue is that the Norwegian linguistic resources do not contain a large range of tasks to evaluate and compare these models on, as opposed to the English benchmark suites like GLUE (Wang et al., 2018), SuperGLUE (Wang et al., 2019) or GLGE (Liu et al., 2021), to name a few.

We present two new datasets for evaluating the understanding large language models have of Norwegian grammar, jointly called the Norwegian corpus of linguistic acceptability (NoCoLA). We hope that the datasets can contribute to the development of a language model benchmark suite for Norwegian. Our work is limited to the most widely used of the written standards for Norwegian, namely Bokmål. This paper proposes two different views on the same set of sentences, each with a slightly different purpose:

```
# Incorrect (inflection):
    Samfunnet ville bli mer fornøyet.
# Correct:
    Samfunnet ville bli mer fornøyd.

# Incorrect (word choice):
    Jeg er ikke nordmann, med jeg trives i Norge.
# Correct:
    Jeg er ikke nordmann, men jeg trives i Norge.
```

Listing 1: Two illustrative examples of incorrect / correct sentence pairs from **NoCoLA**$_{zero}$. The English translations: *"Society would be happier"* and *"I'm not Norwegian, but I enjoy living in Norway."*

- **NoCoLA**$_{class}$ is a collection of sentences split into two classes: grammatically acceptable and non-acceptable. Thus, it is a binary classification task, where a language model is expected to be first fine-tuned on the training data split. This task is more practically-oriented and evaluates the fine-tuning abilities of a language model. The downside is that we cannot tell if the performance comes from its innate abilities or if it was obtained from the supervised fine-tuning.
- **NoCoLA**$_{zero}$ is a collection of pairs of sentences, where only one of them is grammatically acceptable. Here, we do not fine-tune on this task at all, the language model gives a probability to each of the two sentences, and we measure how often the correct one gets a higher probability. While not as practical as the first task, the zero-shot evaluation provides a better estimate of the innate grammatical understanding.

We provide a comprehensive evaluation of the existing Norwegian language models and release the

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 610–617

data and code for an easy evaluation of new Norwegian models.[1]

2 Related work

The closest equivalents of our **NoCoLA**$_{class}$ dataset are the English Corpus of Linguistic Acceptability (CoLA; Warstadt et al., 2019) and the Swedish Dataset for Linguistic Acceptability Judgments (DaLAJ; Volodina et al., 2021). On the other hand, **NoCoLA**$_{zero}$ roughly follows the The Benchmark of Linguistic Minimal Pairs for English and the English (BLiMP; Warstadt et al., 2020).

Data sources. There are two primary strategies for obtaining non-acceptable sentences for a corpus of linguistic acceptability. The non-acceptable sentences are either collected from the linguistics literature by experts as in the English, Russian and Italian corpora (Warstadt et al., 2019; Mikhailov et al., 2022; Trotta et al., 2021) – or these sentences are collected from natural texts, usually based on the language of language learners, such as in the Swedish acceptability corpus (Volodina et al., 2021). The second, natural, approach is also used for the creation of our Norwegian corpus.

CoLA. This dataset consists of 10 600 acceptable and non-acceptable sentences collected manually from the linguistics literature, with the goal of covering specific linguistic phenomena – and the morphological, syntactic and semantic violation of rules connected to those phenomena. By collecting the data in this manner, one ensures that the dataset represents language phenomena that are central to human linguistic competence according to linguists. CoLA has become a standard task for evaluating English language models after it was included in the GLUE benchmark for natural language understanding (Wang et al., 2018). Similar datasets for Russian (RuCoLa; Mikhailov et al., 2022) and Italian (ItaCoLA; Trotta et al., 2021) follow the same methodology as the English CoLA.

DaLAJ. This dataset has the same purpose for benchmarking Swedish language models as CoLA has for English. In contrast to the English CoLA, DaLAJ uses the error-annotated learner corpus SweLL (Volodina et al., 2019) as their source of non-acceptable sentences. DaLAJ contains 4 798 sentence pairs, where the non-acceptable versions are annotated with one of four error-tags. All of the error-types in DaLAJ vol.1 focus on semantic aspects of the sentences, and morphological and syntactic error types are left for future work. The original sentences are edited so that each sentence only has one error focus.

BLiMP. The BLiMP dataset consists of 67 000 minimal pairs, all of them generated artificially. Some examples of phenomena covered in the dataset are determiner-noun agreement, verb argument structure and irregular verb-forms. Each pair differs only on one single parameter, namely the element that leads to the non-acceptability.

Comparison with NoCoLA. Our datasets fill the same purpose for evaluation of language models in Norwegian as CoLA and BLiMP do for English. However, the source of the sentences is different, as we follow the methodology used for DaLAJ. Our data consists of naturally produced sentences, instead of controlled and artificially generated ones. Where CoLA collects sentences that are handpicked by linguists to represent specific linguistic phenomena, our sentences contain errors that mirror the natural distribution of errors in texts by second language learners. Thus, NoCoLA gives an indication of how well a given language model distinguishes between acceptable and non-acceptable Norwegian text, but not of how well it understands the full range of possible grammatical phenomena of the language. NoCoLA is also substantially larger than CoLA, with almost 15 times more examples. The NoCoLA error types are not comparable to BLiMP, where the error-types describe the underlying grammatical problem. Instead, the NoCoLA error-types describe the changes that need to be made to correct the errors. In contrast to DaLAJ we keep original sentences belonging to all error type categories, including morphological, syntactic, and semantic errors.

3 Datasets description

3.1 ASK corpus

Both **NoCoLA**$_{class}$ and **NoCoLA**$_{zero}$ require a source for both acceptable and non-acceptable sentences. The latter is hard to come by in most naturalistic text by adult native speakers. Our source for both NoCoLA datasets is the ASK Corpus – A Language Learner Corpus of Norwegian as a Second Language (Tenfjord et al., 2006). It consists of submissions by second language learners of Norwegian Bokmål around the year 2000, each

[1] https://github.com/ltgoslo/nocola

with one or more essays. The essays are written as solutions to two separate Norwegian language exams, which are estimated in Berggren (2019) to be approximately CEFR-levels B1 and B2.

There are 1 935 submissions, with 46 000 original sentences in total. Each essay has been manually corrected by native speakers, hereby called correctors. The errors in the corpus are annotated with a set of error-codes, which indicate the change that needs to be done to correct the original passage. For instance, "F" indicates wrong morpho-syntactic category, while "PUNCM" means that punctuation is missing, and needs to be added. We have merged some of the error-codes so that we have a medium-grained way of understanding the performance of the models on the different types of errors found in **NoCoLA**$_{zero}$. A short explanation of these error-codes can be found in the appendix.

3.2 Conversion from ASK to NoCoLA

Privacy considerations The original ASK corpus is annotated with rich metadata about the learners. For this dataset we have decided to surpass all this metadata, including the CEFR-level of the test. ASK has also gone through a anonymization process, where possibly sensitive words have been replaced by placeholders. Still, some of the topics of the essays deal with so specific topics about the lives of the learners, that we decided to sentence-scramble the essays to achieve maximum anonymity.

Sentence merging. For the NoCoLA datasets we want sentences as the unit for evaluation. Therefore we need to split the continuous text of ASK into sentences. However, since some of the corrections suggested by the correctors affect the way the text is split into sentences, and we need alignment between the acceptable and non-acceptable in the pairs for **NoCoLA**$_{zero}$, we decided to always keep the longest available version in cases where there is disagreement between both versions. The principle applies to both datasets. Thus, the unit referred to as "sentence" in this paper can consist of multiple sentences.

Error extraction. For each of these sentences, we first extract a corrected (acceptable) version. In order to test only minimal errors and to label each non-acceptable sentence with an error-type, we generate one non-acceptable sentence for each error found in the originals. Therefore we extract almost 100 000 non-acceptable sentences, as many of the original sentences have multiple errors.

Dataset	Train	Dev	Test
NoCoLA$_{class}$	116 195	14 289	14 383
NoCoLA$_{zero}$	—	—	99 115

Table 1: Number of sentences and sentence pairs, respectively, for both NoCoLA datasets.

Post-processing. We did a few additional adjustments to the dataset. All sentences are heuristically detokenized and removed if they contain an uneven count of quotation marks. If no error type is mentioned for a given correction, we also remove that sentence. The sensitive words that have been replaced by placeholders like "@sted" (place) and "@navn" (name) are replaced with a substitute representation of that category, i.e. "Oslo" instead of "@sted", to normalize all sentences. This is to avoid feeding too many unknown tokens to the language models. In rare occasions, these replacements might cause some sentences to become erroneous, since the possible genitive and plural conjugations in the original texts are not annotated with separate placeholder-tokens.

Conversion results. The final dataset contains 144 867 sentences, 31.5% of which are acceptable. **NoCoLA**$_{class}$ has been shuffled and then randomly split to ensure unbiased development and test sentences. The split has been done in an approximate 80:10:10 ratio, resulting in the sentence-level statistics from Table 1.

4 Baseline models

4.1 Evaluation of NoCoLA$_{class}$

In order to evaluate language models on **NoCoLA**$_{class}$, we use the standard fine-tuning approach from Devlin et al. (2019). Accordingly, every sentence is tokenized, prepended by a special [CLS] token, appended by a [SEP] token and input to a pre-trained language model. Subsequently, the contextualized representation of the special [CLS] token is fed into a binary MLP classifier. The pre-trained weights of the language model are further trained together with the classifier weights.

Model	Inflection	Word choice	Spelling	Missing	Superfluous	Punctuation	Word order	Capitalization	Compounding	Derivation	Overall
$\text{BERT}_{\text{base}}$ (Devlin et al., 2019)	50.70	53.55	63.43	60.44	51.69	79.33	51.85	82.54	54.31	54.11	59.48
$\text{mBERT}_{\text{base}}$ (Devlin et al., 2019)	79.92	69.05	90.74	76.91	78.84	83.97	74.88	87.88	78.72	80.44	79.53
$\text{XLM-R}_{\text{base}}$ (Conneau et al., 2020)	91.43	85.28	92.60	87.43	87.56	83.93	84.33	90.60	89.63	91.96	88.02
ScandiBERT (Hofmann et al., 2022)	93.43	89.79	90.84	90.14	90.05	87.10	90.08	90.55	85.82	90.68	90.27
$\text{NB-BERT}_{\text{base}}$ (Kummervold et al., 2021)	93.76	89.19	**97.14**	86.54	**92.48**	73.98	90.94	92.73	**91.15**	**94.70**	89.04
NorBERT_1 (Kutuzov et al., 2021)	93.46	88.46	94.54	88.66	89.41	**88.46**	92.01	**94.26**	90.83	93.05	**90.83**
NorBERT_2 (Kutuzov et al., 2021)	91.66	88.20	96.88	89.22	90.91	75.82	92.67	93.13	74.18	92.69	88.51
$\text{NorBERT}_{3,\text{ base}}$ (Samuel et al., 2023)	94.63	90.98	87.06	91.04	90.23	87.25	89.82	92.73	86.95	89.21	90.44
$\text{XLM-R}_{\text{large}}$ (Conneau et al., 2020)	92.54	88.17	90.06	88.57	89.28	80.84	84.52	91.35	89.70	93.24	88.27
$\text{NB-BERT}_{\text{large}}$ (Kummervold et al., 2021)	**95.20**	**92.41**	95.16	**91.47**	91.92	85.33	**93.36**	17.01	89.56	92.87	90.51
$\text{NorBERT}_{3,\text{ large}}$ (Samuel et al., 2023)	94.89	91.98	83.71	**91.47**	90.84	86.39	87.87	92.48	84.19	88.30	90.01

Table 2: The accuracy values of zero-shot evaluation on **NoCoLA**$_{zero}$. Fine-grained results over different error types are reported (Appendix A), as well as the overall average over all sentence pairs in the datasets.

4.2 Evaluation of NoCoLA$_{zero}$

One disadvantage of **NoCoLA**$_{class}$ is that the results are skewed by the second-stage supervised training and it can be problematic to disentangle the properties of the LM from the classifier (Belinkov, 2022). In contrast, pure LM-based evaluation of **NoCoLA**$_{zero}$ attempts to measure the linguistic knowledge of a language model in a zero-shot manner – without any additional training. The dataset consists of 99 115 sentence pairs; each pair differs minimally on the surface level, but only one of the sentences is acceptable. We can use the intrinsic ability of language models to assign a probability to every sentence and test how often a language model assigns a higher probability to the correct sentence, as in (Warstadt et al., 2020). As the two classes are perfectly balanced, simple accuracy is a sufficient metric for this setup.

CLM evaluation. The *causal* language models are trained to estimate $p(\boldsymbol{s}_t|\boldsymbol{s}_{<t})$ for sentence $\boldsymbol{s}$ and token $\boldsymbol{s}_t$ where $\boldsymbol{s}_{<t} = (\boldsymbol{s}_i|i < t)$; then the sentence log-probability is simply given by $\log p(\boldsymbol{s}) = \sum_{t=1}^{N} \log p(\boldsymbol{s}_t|\boldsymbol{s}_{<t})$.

MLM evaluation. The issue with *masked* language models is that they are not designed to calculate the joint probability; they are trained to estimate $p(\boldsymbol{s}_t|\boldsymbol{s}_{\setminus t})$ – the likelihood of a token s_t given its bidirectional context $\boldsymbol{s}_{\setminus t} = (\boldsymbol{s}_i|i \neq t)$. We can however still use MLMs to infer a *score* for each sentence where a higher *score* corresponds to a more likely sentence. Wang and Cho (2019) defined *pseudo-log-likelihood score* of a sentence s with model θ as

$$\text{PLL}(\boldsymbol{s}) = \frac{1}{N}\sum_{t=1}^{N} \log p(\boldsymbol{s}_t|\boldsymbol{s}_{\setminus t}; \theta).$$

Salazar et al. (2020) tested PLL and found that it produces accurate predictions on BLiMP. We adopt

Model	Lang.	Size	Accuracy	MCC
$\text{BERT}_{\text{base}}$	en	110M	$69.56^{\pm 0.37}$	$23.99^{\pm 0.41}$
$\text{mBERT}_{\text{base}}$	multi	178M	$75.28^{\pm 0.66}$	$46.39^{\pm 0.67}$
$\text{XLM-R}_{\text{base}}$	multi	278M	$79.29^{\pm 0.20}$	$55.14^{\pm 0.36}$
ScandiBERT	multi	124M	$80.25^{\pm 0.33}$	$57.12^{\pm 0.37}$
$\text{NB-BERT}_{\text{base}}$	no	178M	$80.69^{\pm 0.44}$	$58.10^{\pm 0.48}$
NorBERT 1	no	111M	$71.53^{\pm 0.80}$	$35.85^{\pm 1.70}$
NorBERT 2	no	125M	$79.99^{\pm 0.27}$	$56.09^{\pm 0.30}$
$\text{NorBERT}_{3,\text{ base}}$	no	123M	$81.50^{\pm 0.16}$	$59.21^{\pm 0.28}$
$\text{XLM-R}_{\text{large}}$	multi	560M	$81.03^{\pm 0.27}$	$58.56^{\pm 0.30}$
$\text{NB-BERT}_{\text{large}}$	no	355M	$81.43^{\pm 0.32}$	$59.68^{\pm 0.14}$
$\text{NorBERT}_{3,\text{ large}}$	no	353M	$\mathbf{82.48}^{\pm 0.21}$	$\mathbf{60.96}^{\pm 0.45}$

Table 3: Accuracy and the Matthews correlation coefficient (Matthews, 1975), the main metric of **NoCoLA**$_{class}$. We report the mean and standard deviation across five runs on the test split.

their approach and evaluate our models with PLL.

5 Results

5.1 Results on NoCoLA$_{class}$

The results from benchmarking the publicly available Norwegian language models on the classification task can be seen in Table 3. The classification accuracy is around 80% for for these models. One exception is the slightly older NorBERT 1, which performs substantially worse, even if being trained on clean Norwegian data: Wikipedia and newspaper articles (Kutuzov et al., 2021). We use the English BERT$_{\text{base}}$ as a naive baseline, which gives us a lower bound on the performance of any decent Norwegian language models. The three largest models give a small increase in performance compared to the base-sized versions of the same models. The NorBERT$_3$ models (Samuel et al., 2023) consistently outperform other models on this task.

5.2 Results on NoCoLA$_{zero}$

On the raw zero-shot diagnostic task (Table 2), all models trained on Norwegian or Scandinavian languages perform well with results around 90% accuracy. The best performance comes, perhaps surprisingly, from NorBERT 1 – possibly because it was pre-trained on a relatively small and clean corpus. Remarkably, increased number of parameters does not seem to improve performance on this task.

We have also included accuracy scores for the individual error-types, as these fine-grained scores can be used as a helpful cue for NLP researchers who develop new language models. Comparably low scores can signal a problem with their training corpus or with their tokenizer. For example, the two NB-BERT models are relatively weak on punctuation-related errors. The large version is trained on uncased data, which explains this models inability to understand the case-related errors. ScandiBERT performs comparably to the Norwegian ones on most parameters except for spelling.

6 Conclusion

In this paper we have proposed NoCoLA, the first dataset for linguistic acceptance in Norwegian Bokmål. We showed how to use it for measuring the linguistic knowledge of language models on both a classification task and a zero-shot probability comparison task. We have described how the datasets were created and what their motivation is, compared them to related work in English NLP and showed how to use them for fine-grained error analysis of language models.

Lastly, we evaluated existing Norwegian masked language models on both proposed tasks. These results suggest that models trained specifically for Norwegian or Scandinavian languages perform better at discriminating between acceptable an non-acceptable sentences. The classification results also show that linguistic acceptability is a relatively hard task, as only one of the models achieved more than 60% on the main MCC metric. The results on our diagnostic dataset highlight some shortcoming of the existing models. We will release all evaluation sources in the camera-ready version.

References

Yonatan Belinkov. 2022. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219.

Stig Johan Berggren. 2019. Automated assessment of norwegian l2 essays using multi-task learning. master thesis, university of oslo.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Valentin Hofmann, Goran Glavaš, Nikola Ljubešić, Janet B. Pierrehumbert, and Hinrich Schütze. 2022. Geographic adaptation of pretrained language models.

Per E Kummervold, Javier De la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. 2021. Operationalizing a national digital library: The case for a Norwegian transformer model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 20–29, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-scale contextualised language modelling for Norwegian. In

Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa), pages 30–40, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, Pengcheng Wang, Jiusheng Chen, Daxin Jiang, Jiancheng Lv, Ruofei Zhang, Winnie Wu, Ming Zhou, and Nan Duan. 2021. GLGE: A new general language generation evaluation benchmark. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 408–420, Online. Association for Computational Linguistics.

B.W. Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451.

Vladislav Mikhailov, Tatiana Shamardina, Max Ryabinin, Alena Pestova, Ivan Smurov, and Ekaterina Artemova. 2022. RuCoLA: Russian corpus of linguistic acceptability. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5207–5227, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

David Samuel, Andrey Kutuzov, Samia Touileb, Erik Velldal, Lilja Øvrelid, Egil Rønningstad, Elina Sigdel, and Anna Sergeevna Palatkina. 2023. Norbench – a benchmark for norwegian language models. In *The 24rd Nordic Conference on Computational Linguistics*.

Kari Tenfjord, Paul Meurer, and Knut Hofland. 2006. In *The ASK Corpus – A Language Learner Corpus of Norwegian as a Second Language. Proceedings from 5th International Conference on Language Resources and Evaluation (LREC), Genova 2006.* [link].

Daniela Trotta, Raffaele Guarasci, Elisa Leonardelli, and Sara Tonelli. 2021. Monolingual and cross-lingual acceptability judgments with the Italian CoLA corpus. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2929–2940, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Elena Volodina, Lena Granstedt, Arild Matsson, Beáta Megyesi, Ildikó Pilán, Julia Prentice [Grosse], Dan Rosén, Lisa Rudebeck, Carl-Johan Schenström, Gunlög Sundberg, and Mats Wirén. 2019. The swell language learner corpus: From design to annotation. *The Northern European Journal of Language Technology*, 6:67–104.

Elena Volodina, Yousuf Ali Mohammed, and Julia Klezl. 2021. Dalaj - a dataset for linguistic acceptability judgments for swedish: Format, baseline, sharing. *CoRR*, abs/2105.06681.

Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis, Minnesota. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

A NoCoLA$_{zero}$ error types[2]

- **Inflection:** wrong form of word. Merged from ASK-codes "F": wrong morpho-syntactic form and "INFL": suffix from correct category, but wrong form for this particular word. *"Jeg vet ikke hvorfor jeg har valgt **dette** oppgaven." "I do not know why I have chosen this task."*
- **Word choice:** wrong choice of word. Merged from ASK-codes "W": wrong word and "FL": word from another language. *"Jeg er et eksempel **for** det." "I am an example of that"*
- **Spelling:** wrong spelling of word, corresponding to ASK-code "ORT". *"De er en rik **fammilie**." "They are a rich family."*
- **Missing:** word should be added. Corresponding to ASK-code "M". *"Norge kan bidra veldig mye på Europeiske planet." "Norway can contribute a lot at **the** European level."*
- **Superfluous:** word should be removed. Corresponding to ASK-code "R". *"Da mistet jeg den beste vennen **min** i hele livet mitt." "Then I lost the best friend in my whole life."*
- **Punctuation:** add or remove punctuation. Corresponding to ASK-codes "PUNC", "PUNCM" and "PUNCR". *"Hva skal jeg gjøre etterpå**.**" "What should we do afterwards?"*
- **Word order:** wrong order of words or phrases. Corresponding to ASK-code "O". *"Hvis du har tillatelse, **du kan** fiske også." "If you have a licence, you can fish as well."*
- **Capitalization:** add/remove capitalization. Corresponding to ASK-code "CAP". *"**nå** liker jeg meg godt i Oslo." "Now I enjoy myself in Oslo"*
- **Compounding:** deviation regarding compounding. Corresponding to ASK-codes "PART" and "SPL". *"**Etter på** skal jeg studere for å bli sykepleier." "Afterwards I want to study to become a nurse."*
- **Derivation:** deviation regarding derivation. Corresponding to ASK-code "DER". *"Derfor er jeg helt enig med **forbudelse** mot krenkende uttalelser." "Therefore I completely agree with the ban on offensive statements."*
- **Other:** any other error.

[2]The codebook given to the correctors of ASK contains a multitude of additional tags which are not used by the correctors. For example there is OINV for "subject/verb inversion i contexts where there should not be one".

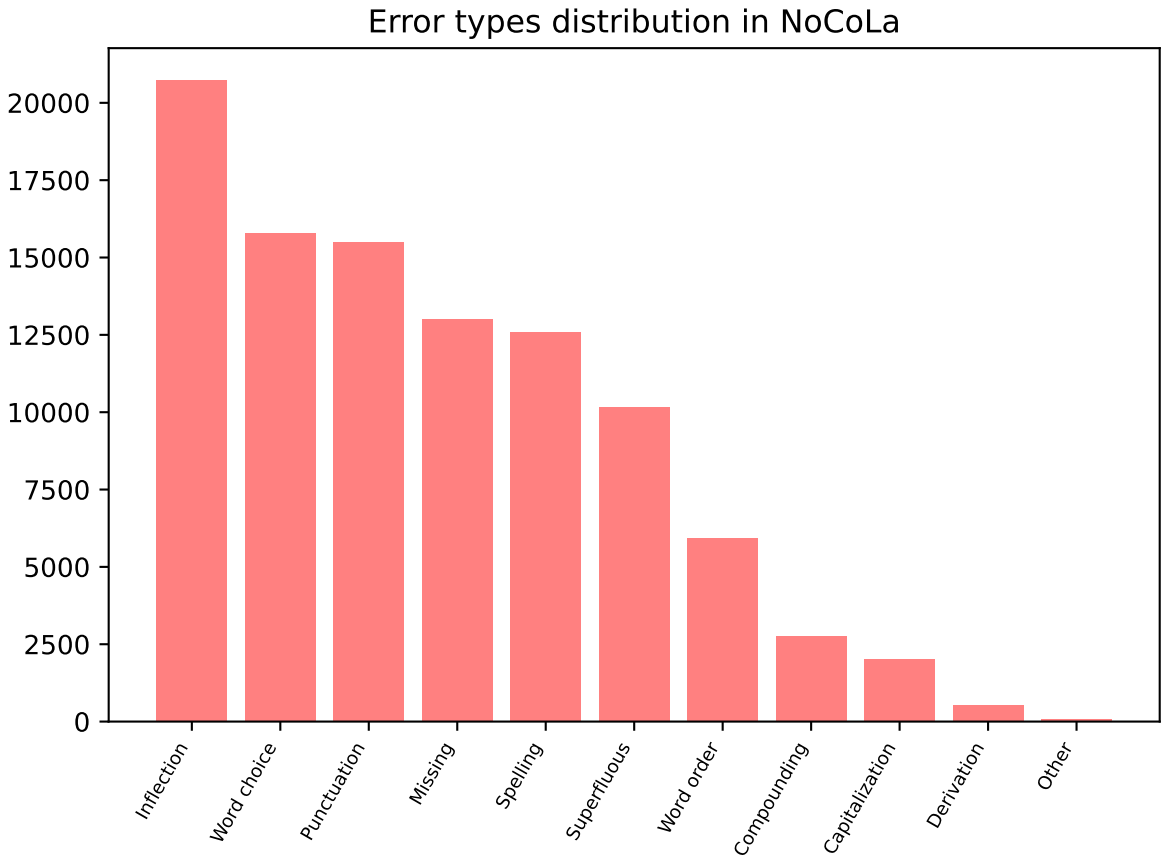


Figure 1: Distribution of error types in the NoCoLA datasets.

NorBench – A Benchmark for Norwegian Language Models

David Samuel,[1] **Andrey Kutuzov,**[1] **Samia Touileb,**[2] **Erik Velldal,**[1] **Lilja Øvrelid,**[1]
Egil Rønningstad,[1] **Elina Sigdel**[1] **and Anna Palatkina**[1]
[1]University of Oslo, Language Technology Group
[2]University of Bergen, MediaFutures

Abstract

We present NorBench: a streamlined suite of NLP tasks and probes for evaluating Norwegian language models (LMs) on standardized data splits and evaluation metrics. We also introduce a range of new Norwegian language models (both encoder and encoder-decoder based). Finally, we compare and analyze their performance, along with other existing LMs, across the different benchmark tests of NorBench.

1 Introduction

This paper provides a suite of standardized tasks and probes for benchmarking of Norwegian language models (LMs). In addition to collecting a broad range of annotated datasets, we provide precise task definitions, pre-defined data splits and evaluation metrics, with corresponding scripts for streamlining the entire benchmarking pipeline. The resulting resource is dubbed NorBench. We furthermore present a range of new transformer-based language models (LMs) for Norwegian, trained with optimized configurations and architectures, and on different corpora with different pre-processing. Our contributions are as follows:

1. We introduce NorBench, a collection of Norwegian datasets and evaluation scripts that ensures simple, fair and standardized comparison between Norwegian LMs. The existing models from prior work are evaluated and compared. All data and code related to NorBench are publicly available online.[1]

2. An integral part of NorBench is diagnostic set of tasks that probe the affinity of LMs towards gender-bias and toxic language – an unfortunate side-effect for many models pre-trained on large amounts of text.

[1]https://github.com/ltgoslo/norbench

Task	Train	Dev	Test
Morpho-syntactic token-level tasks			
Tokens in UD tasks	489 217	67 619	54 739
Named entities	23 071	2 942	2 393
Sentiment analysis			
SA documents	34 903	4 360	4 351
SA sentences	7 973	1 411	1 181
SA targets	5 044	877	735
Linguistic acceptability			
NoCoLA sentences	116 195	14 289	14 383
Question answering			
NorQuAD questions	3 808	472	472
Machine translation			
Bokmål–Nynorsk sentences	10 000	10 000	10 000

Table 1: Number of labeled entities in the training, development, and test splits in the datasets used for the NorBench tasks.

3. We develop a new generation of Norwegian LMs – NorBERT$_3$ and NorT5 – that achieve state-of-the-art performance across most NorBench tasks. We provide multiple sizes of these models and show that even small versions maintain competitive performance.

4. We empirically test the impact of different available Norwegian training corpora on the downstream performance. Our results suggest that pre-training on a simple concatenation of all available resources is not always beneficial.

The rest of the paper is structured as follows. Section 2 provides an overview of the tasks included in NorBench. In Section 3, the details of our evaluation workflow are outlined. The architecture and the training collections of our novel LMs are described in Section 4, while in Section 5, we summarize and analyze the benchmarking results. Section 6 briefly describes prior work, while we point out directions for future work in Section 7, before concluding in Section 8.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 618–633

2 NorBench task descriptions

We here briefly describe each task and associated dataset. The number of training examples for the different datasets and their train, development, and test splits are provided in Table 1. For the full details about each task, we refer the reader to the NorBench GitHub repository. Before we describe the various tasks, we first briefly comment on the two official varieties of written Norwegian.

Bokmål and Nynorsk Norwegian has two official written standards: *Bokmål* (BM), used by 85–90% of the Norwegian population, and *Nynorsk* (NN). While they are closely related, there can be relatively large lexical differences. The contextualised LMs presented in this paper are therefore trained jointly on both varieties, but with the minority variant Nynorsk represented by comparatively less data than Bokmål (reflecting the natural usage). Several previous studies have indicated that joint modeling of Bokmål and Nynorsk works well for many NLP tasks, like tagging and parsing (Velldal et al., 2017) and NER (Jørgensen et al., 2020). In cases where the labeled data for our benchmark tasks are available as separate versions for Bokmål and Nynorsk, we fine-tune models jointly on the combined BM/NN data. One practical advantage of training joint and 'variety agnostic' models, is that only a single model needs to be maintained, and we bypass the need for a separate 'language identification' step.

2.1 Morpho-syntactic token-level tasks

UD tasks We use the Norwegian Universal Dependencies Treebank (Øvrelid and Hohle, 2016; Velldal et al., 2017) from UD 2.11,[2] in turn converted from NDT (Solberg et al., 2014). In order to evaluate the general performance on Norwegian, we concatenate the Bokmål (BM) and Nynorsk (NN) datasets for both fine-tuning and evaluation. The models are challenged to predict universal part-of-speech tags (UPOS), universal features (UFeats), lemmas and dependency trees (Nivre et al., 2016).

UPOS tags cover the basic POS categories (17 tags) and *UFeats* differentiate more fine-grained lexical and grammatical properties of words, e.g. number, gender, and tense (172 tags in total). Both tagging tasks use the standard accuracy metric.

Lemmatization evaluates how well a language model understands Norwegian morphology and in order to transform an inflected word into its lemmatized form. An integral part of lemmatization – in our variety-agnostic setting – is implicit language identification, because Bokmål and Nynorsk have different lemmatization standards. Correct prediction requires exact match with the gold lemma; we report the aggregated token-wise accuracy.

Dependency parsing involves identifying the relationships between words in a sentence, resulting in a dependency tree that represents the grammatical structure of the sentence. By evaluating the quality of dependency parsing outputs by a language model, one can determine its ability to recognize and categorize the grammatical roles of words based on their syntactic function. We report the labeled attachment score (LAS), the standard evaluation metric for dependency parsing.[3]

Named entity recognition We use the NorNE[4] dataset which annotates the UD/NDT (for both Bokmål and Nynorsk) with named entities (Jørgensen et al., 2020). We predict 8 entity types: Person (PER), Organization (ORG), Location (LOC), Geo-political entity, with a locative sense (GPE-LOC), Geo-political entity, with an organization sense (GPE-ORG), Product (PROD), Event (EVT), and Nominals derived from names (DRV). The evaluation metric used is 'strict' micro F_1, requiring both the correct entity type and exact match of boundary surface string. It is computed using the code for the SemEval'13 Task 9.[5]

2.2 Sentiment analysis

Document-level ternary polarity classification The Norwegian Review Corpus (NoReC; 2nd release) (Velldal et al., 2018) comprises 43 425 professional reviews from a range of Norwegian news sources, and covering a range of different domains (e.g., books, movies, games, music, various consumer goods, etc.). The average length of a document is 389 words. While the reviews originally come with numerical ratings on a scale of 1–6, we here conflate these to three classes; *negative* (ratings 1–3), *fair* (4), and *positive* (5–6). This mapping is done to avoid problems with too few examples for the ratings in the extreme ends of the numerical scale. The dataset comes with prede-

[2] http://hdl.handle.net/11234/1-4923

[3] https://universaldependencies.org/conll18/evaluation.html

[4] https://github.com/ltgoslo/norne

[5] https://github.com/davidsbatista/NER-Evaluation

fined data splits (chronologically sorted), and we evaluate using macro F_1.

Sentence-level ternary sentiment classification We include the dataset NoReC$_{sentence}$[6] for training and evaluating on the task of sentence-level polarity classification with respect to three classes (positive, negative, or neutral). As described by Kutuzov et al. (2021), this data is derived from NoReC$_{fine}$ (Øvrelid et al., 2020), a subset of NoReC, by aggregating the fine-grained annotations to the sentence-level, removing sentences with mixed sentiment. The evaluation metric is macro F_1.

Targeted sentiment analysis We use the NoReC$_{tsa}$[7] dataset for the task of targeted sentiment analysis (TSA). As described in Rønningstad et al. (2022), the data is derived from NoReC$_{fine}$ by only including target expressions and the associated positive/negative polarity. The task is to jointly predict the target spans and their polarity, and we use the same evaluation strategy as for NER.

2.3 Linguistic acceptance

NoCoLA Norwegian corpus of linguistic acceptance (NoCoLA; Jentoft and Samuel, 2023) is used to evaluate language models on their understanding of Norwegian grammaticality. NoCoLA is derived from the ASK Corpus – a language learner corpus of Norwegian as a second language (Tenfjord et al., 2006), which contains texts written exclusively in Norwegian Bokmål, not covering the Nynorsk variety. We report the Matthews correlation coefficient (MCC; Matthews, 1975) on NoCoLA$_{class}$, the official binary sentence classification variant of the dataset.

2.4 Question answering

NorQuAD is a Norwegian extractive question answering dataset which consists of 4 752 manually created question-answer pairs based on Wikipedia and news articles (Ivanova et al., 2023).[8] We here report token-level F1; human performance on the test portion of the dataset has been measured at 91.1% F1 (Ivanova et al., 2023).

2.5 Machine translation

Bokmål–Nynorsk translation. The fact that a monolingual Norwegian language model is actually trained on two language varieties – Bokmål and Nynorsk – allows us to evaluate generative models on machine translation. We collect the available Bokmål–Nynorsk bitexts,[9] deduplicate the sentences on both sides and split them into training, development, and test portions, each with 10 000 parallel sentences. We evaluate the translation from Bokmål to Nynorsk using SacreBLEU (Lin and Och, 2004; Post, 2018).[10]

2.6 Diagnostics of harmful predictions

Unlike the previous items, this is not a 'task', but rather a description of important model properties. We follow previous works on Norwegian to probe our language models for gender bias in occupations, as well as assessing the harmfulness of their sentence completions (Touileb et al., 2023, 2022; Touileb and Nozza, 2022).

3 NorBench baseline methodology

Below we describe various choices pertaining to fine-tuning the LMs for the various tasks. Note that, all of the approaches described here should be considered baselines, in the sense that the goal is not to produce state-of-the-art results, but rather to implement simple evaluation approaches allowing for a fair comparison of different LMs across the various tasks.

3.1 A joint model for UD tasks

Since the UD tasks are annotated within the same dataset, we evaluate them jointly with a single multi-task model. We follow the multi-task setup from UDify (Kondratyuk and Straka, 2019): first, we take a separate weighted convex combination of hidden layers for every subtask. Then, we average-pool these contextualized subword representations to get a vector embedding for each token. Finally, these vectors are input to classification heads for UPOS and UFeats tagging, to a classification head for predicting lemma transformation rules, and to biaffine attention heads for dependency parsing (Dozat and Manning, 2017).

[6] https://github.com/ltgoslo/norec_sentence

[7] https://github.com/ltgoslo/norec_tsa

[8] https://github.com/ltgoslo/NorQuAD

[9] Provided by the National Library of Norway: https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-78/, https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-47/

[10] The SacreBLEU metric involves several parameters that change the outcomes (Post, 2018), we use BLEU with no smoothing, `13a` tokenization and no lowercasing – the default values in `torchmetrics 0.11.4`: https://torchmetrics.readthedocs.io/en/stable/text/sacre_bleu_score.

3.2 Text classification

For document- and sentence-level sentiment analysis, together with classification of linguistic acceptability, we utilize the same text classification approach, based on the widely-used fine-tuning scheme from Devlin et al. (2019). There, every tokenized text sequence is prefixed by a special `[CLS]` token, appended by a `[SEP]` token and passed into a pre-trained language model, which produces a contextualized representation for the special `[CLS]` token. Finally, this representation is passed into the downstream classifier that produces the final prediction among the available classes. For the encoder-decoder models, we chose three target words as the class labels ('negativ', 'nøytral' and 'positivt') and fine-tuned the models to generate these target words given the input text. At the inference time, an input text is assigned a class depending on whether the corresponding target word occurs in the generated text.

3.3 Sequence labeling for NER and TSA

NER and TSA are approached as a sequence labeling task where we classify text spans by tagging tokens with beginning-inside-outside tags (BIO; Ramshaw and Marcus, 1995).

3.4 Question answering

We follow the SQuAD fine-tuning method introduced in BERT (Devlin et al., 2019). For every question and context passage, the goal is to identify the answer within the passage. The question and passage texts are concatenated together and the evaluated model is trained to predict the first and last token of the answer – the problem is cast as 2-task binary classification problem.

3.5 Machine translation

We use this task only for evaluation of generative sequence-to-sequence models such as T5s (Raffel et al., 2020). This task naturally fits these models – the source sentence is encoded and the target sentence is decoded with the respective parts of the model. We use simple greedy decoding for generation during inference.

3.6 Probing for gender-bias and harmfulness

We take advantage of the MLM objective of the models, and create templates consisting of gendered head-words, followed by predicates.

Gender-bias To probe for gender bias in occupations, we follow and use the templates of Touileb et al. (2023) and Touileb et al. (2022). These templates are sequences of masked gendered head-words (*e.g.* the woman, the man, the sisters, the uncles ...), followed by predicates pertaining to verbs related to performing an occupation (*e.g.* is, was, worked as, ...), followed by a set of occupations extracted from the Norwegian Statistics bureau (Touileb et al., 2022). Using the probabilities of the masked gendered head-words, we compute the two bias scores: normative and descriptive as defined by Touileb et al. (2023). The normative score compares the gender-based aggregated probability distributions of templates with a normative distribution of genders in occupations. The idea here is that genders should be equally represented in each occupation with a gender distribution falling between 45% and 55% for each occupation. The descriptive bias score compares the probability distribution of genders across occupations as represented in language models, to the real world distribution of these genders based on the Norwegian Statistics bureau data.

Harmfulness We also follow Touileb and Nozza (2022) to assess the harmfulness of sentence-completions of each language model. We use their templates, constructed similarly to the previous templates where the head-words are gendered-nouns followed by predicates as defined by Nozza et al. (2021), and where the last token is masked. The probing is therefore aiming at completing sentences, by looking at top one, five, ten, and twenty most likely words for each template. Once the completions returned by the models, we compute the HONEST score (Nozza et al., 2021). This score is a word-level completion score that maps generated completions to their respective language-specific HurtLex (Bassignana et al., 2018) lexicon of offensive words. The scores represent the total number of completions existing in the lexicon compared to the total amount of returned completions.

4 New Norwegian language models

A number of large Norwegian language models have appeared in recent years: to name only the masked LMs, Kutuzov et al. (2021) trained NorBERT_1, followed by NorBERT_2, and Kummervold et al. (2021) introduced NB-BERT models, coming in different sizes. In this paper, we present a set of novel masked and text-to-text LMs for

Norwegian trained according to the LTG-BERT training recipe by Samuel et al. (2023). We dub these models **NorBERT$_3$** and **NorT5** and evaluate their performance across different model sizes and training corpora.

4.1 Training corpora

Text sources Our LM training dataset included the following text collections:

- Norwegian Wikipedia dumps (BM/NN) from October 2022; about 180 million words;
- NBDigital, public domain texts released by the National Library (NB) of Norway in 2015; 660 million words;[11]
- Norwegian News Corpus (NAK): a collection of Norwegian news texts (both Bokmål and Nynorsk) published between 1998 and 2019; 1.7 billion words;[12]
- Norwegian Colossal Corpus (NCC): the public part of the large and heterogenous corpus released by NB in 2022[13] (Kummervold et al., 2021); about 6.9 billion words;
- Norwegian part of web-crawled mC4 corpus (Xue et al., 2021); about 15 billion words.

The 'standard' models were trained on the concatenation of these corpora, yielding a training collection of about 25 billion word tokens. In Section 5.2, we investigate the effects of 'oversampling' higher-quality sources and training separate models from scratch on NAK, NCC, mC4, Wikipedia, and NBDigital.

Deduplication Before training, all the corpora were de-duplicated on the paragraph-level, using SimHash[14] and removing exact duplicates. The same was done across corpora, reducing their size up to 10%, depending on the corpus.

Filtering Since the largest portion of our training corpus is sourced from web-crawled texts, it is crucial to filter out any unnatural language. Even though our main web-text source is the multilingual Colossal *Clean* Crawled Corpus (mC4), it still contains noisy texts (Dodge et al., 2021), which was also apparent when we manually investigated some of the Norwegian samples. We follow the filtering heuristics implemented for the MassiveText corpus (Rae et al., 2021) and adapt them for Norwegian.

[11] https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-34/

[12] https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-4/

[13] https://huggingface.co/datasets/NbAiLab/NCC

[14] https://github.com/ChenghaoMou/text-dedup

Hyperparameter	x-small	small	base	large
Number of parameters	15M	40M	123M	353M
Number of layers	12	12	12	24
Hidden dimension	192	384	768	1 024
Attention heads	3	6	12	16

Table 2: The main hyperparameters of our four configurations of NorBERT$_3$ language models. Full list of hyperparameters is given in Table 9.

4.2 Architecture

We employ the masked language modeling approach for pre-training NorBERT$_3$ language models and follow the optimized training method from Samuel et al. (2023). This approach differs from the standard BERT (Devlin et al., 2019) training as follows:

1. Liu et al. (2019) found out that BERT is undertrained and the next-sentence prediction task is unnecessary – we thus pre-train for $8\times$ more steps, use sequence length of 512 throughout the whole training, and utilize only the masked language modeling task (MLM) without next-sentence prediction.
2. SpanBERT (Joshi et al., 2020) and T5 (Raffel et al., 2020) demonstrated the advantages of masking random spans instead of individual subwords as in Devlin et al. (2019). Thus, for our MLM objective, the data loader iteratively samples random spans until 15% of the input text is masked. The length of each span is sampled from Geo(p), where $p = 1/3$.
3. Samuel et al. (2023) compared various configurations of transformer architectures and of the training hyperparameters. We employ the best performing setting for our pre-training. Crucial upgrades involve using the NormFormer layer normalization (Shleifer and Ott, 2022), disentangled attention with relative positions (He et al., 2021) and increased amount of weight decay. Please refer to Samuel et al. (2023) for more pre-training details.

Parameter count We train four LMs of different sizes (Table 2), accommodating users with vary-

ing degrees of computational resources, and to establish a baseline performance across LMs with different number of parameters.

Vocabulary and tokenizer We utilize WordPiece subword tokenizer (Wu et al., 2016) and set its vocabulary size to 50 000. Following GPT-2 (Radford et al., 2019), we represent the text as a sequence of UTF-8 bytes instead of Unicode characters, which substantially reduces the number of out-of-vocabulary tokens. We train the tokenizer on the full corpus utilizing the open implementation from the `tokenizers` library.[15]

NorT5 Some NLP tasks, for example machine translation, require a generative language model. Thus we extend the encoder-only architecture of NorBERT$_3$ to full encoder-decoder transformer and pre-train the resulting model, dubbed NorT5, on text-to-text masked language modeling (T5; Raffel et al., 2020). We use the same text corpus, tokenizer and training settings as in NorBERT$_3$ when applicable. For the T5-specific training choices, we follow T5 version 1.1 – i.e. pre-training only on self-supervised masked LM and no parameter sharing between the embedding and classifier layer.[16]

4.3 Pre-training details

In order to reduce training time, pre-training is parallelized over multiple GPUs with the global batch size of 8 192. The number of GPUs used depends on the size of pre-trained language models, ranging between 16 and 512 AMD Instinct MI250X GPUs, each with 128GB memory. The amount of training steps is 250 000, increasing the training budget of the original BERT models 8 times. NorBERT$_{3,\text{ base}}$ was pre-trained in 280 hours using this setting.

5 Benchmarking results

In addition to our NorBERT$_3$ models, we also benchmark these existing models:

- *BERT* (Devlin et al., 2019): to get a baseline performance, we include the scores of an *English*-only language model. Its scores suggest how much information can be inferred from the supervised datasets without any understanding of Norwegian.
- *mBERT* (Devlin et al., 2019): multilingual BERT pre-trained on 104 languages, including Norwegian. The training was done exclusively on Wikipedia dumps with oversampled texts from lower resource languages.
- *XLM-R* (Conneau et al., 2020): more advanced multilingual LM that outperformed mBERT on most tasks. XLM-R models were trained on CommonCrawl data for 100 languages.
- *NB-BERT* (Kummervold et al., 2021): NB-BERT$_{\text{base}}$ model utilized a warm start from pre-trained mBERT. It was later followed by NB-BERT$_{\text{large}}$ trained from scratch on Norwegian data. Both models are trained on the full – i.e., partially non-public – NCC corpus.
- *NorBERT$_1$ and NorBERT$_2$* (Kutuzov et al., 2021): both models follow the pre-training approach of the original BERT model (Devlin et al., 2019). NorBERT$_1$ is pre-trained on NAK and dumps from both Norwegian Wikipedias, and NorBERT$_2$ utilizes the Norwegian part of mC4 and the public part of NCC.
- *ScandiBERT*: Scandinavian BERT trained on a combination of Danish, Faroese, Icelandic, Norwegian, and Swedish texts. However, more than 60% of the training corpus consists of texts from the Norwegian NCC.[17]

Our NorT5 models are compared with the multilingual T5 models (mT5; Xue et al., 2021) and with a set of so-called North-T5 models – mT5 models further fine-tuned solely on Norwegian (published online in 2022).[18]

5.1 Comparison of models

Table 3 and Table 4 show results across all the current NorBench tasks for all language models described above (sorted by their size in the number of parameters). Note that we deliberately do not report any average score across all tasks, since we believe that such aggregated scores do not contribute to real understanding of strong and weak sides of different models: one should pay attention to the performance in particular tasks of interest.

Encoder-only scores Not surprisingly, one can see that it is the largest monolingual models that tend to perform best across the board, and

[15] https://github.com/huggingface/tokenizers

[16] https://github.com/google-research/text-to-text-transfer-transformer/blob/main/released_checkpoints.md#t511

[17] The training procedure is briefly described here; https://huggingface.co/vesteinn/ScandiBERT.

[18] https://huggingface.co/north/t5_base_NCC

Model	Size	UPOS	UFeats	Lemma	LAS	NER	Doc. SA	Sent. SA	TSA	NoCoLA	NorQuAD
NorBERT$_{3,\text{ x-small}}$	15M	**98.8**$^{\pm0.1}$	**97.0**$^{\pm0.1}$	**97.6**$^{\pm0.1}$	**92.2**$^{\pm0.1}$	**86.3**$^{\pm0.4}$	**69.6**$^{\pm2.4}$	**66.2**$^{\pm1.2}$	**43.2**$^{\pm0.5}$	**47.1**$^{\pm0.5}$	**65.6**$^{\pm3.9}$
NorBERT$_{3,\text{ small}}$	40M	**98.9**$^{\pm0.0}$	**97.9**$^{\pm0.0}$	**98.3**$^{\pm0.1}$	**93.7**$^{\pm0.0}$	**89.0**$^{\pm0.3}$	**74.4**$^{\pm0.5}$	**71.9**$^{\pm1.3}$	**48.9**$^{\pm0.9}$	**55.9**$^{\pm0.2}$	**80.5**$^{\pm1.2}$
BERT$_{\text{base, cased}}$	111M	97.9$^{\pm0.0}$	96.4$^{\pm0.1}$	97.9$^{\pm0.0}$	89.8$^{\pm0.2}$	73.4$^{\pm0.7}$	57.3$^{\pm1.4}$	53.0$^{\pm1.1}$	23.2$^{\pm2.2}$	23.9$^{\pm0.4}$	44.9$^{\pm2.2}$
NorBERT$_1$	111M	98.8$^{\pm0.0}$	97.8$^{\pm0.0}$	98.5$^{\pm0.0}$	93.3$^{\pm0.1}$	86.9$^{\pm0.9}$	70.1$^{\pm0.4}$	70.7$^{\pm0.9}$	45.4$^{\pm1.1}$	35.9$^{\pm1.7}$	72.5$^{\pm1.6}$
NorBERT$_{3,\text{ base}}$	123M	**99.0**$^{\pm0.0}$	**98.3**$^{\pm0.1}$	98.8$^{\pm0.0}$	**94.2**$^{\pm0.1}$	**89.4**$^{\pm0.9}$	**76.2**$^{\pm0.8}$	**74.4**$^{\pm0.3}$	**50.2**$^{\pm0.7}$	**59.2**$^{\pm0.3}$	**86.2**$^{\pm0.3}$
NorBERT$_2$	125M	98.7$^{\pm0.0}$	97.6$^{\pm0.0}$	98.2$^{\pm0.0}$	93.4$^{\pm0.1}$	85.0$^{\pm0.9}$	73.5$^{\pm1.1}$	72.5$^{\pm1.5}$	45.4$^{\pm1.1}$	56.1$^{\pm0.3}$	76.6$^{\pm0.7}$
ScandiBERT	124M	98.9$^{\pm0.0}$	98.1$^{\pm0.0}$	98.7$^{\pm0.0}$	**94.1**$^{\pm0.1}$	**89.4**$^{\pm0.5}$	73.9$^{\pm0.4}$	71.6$^{\pm1.3}$	48.8$^{\pm1.0}$	57.1$^{\pm0.4}$	79.0$^{\pm0.7}$
NB-BERT$_{\text{base}}$	178M	98.9$^{\pm0.0}$	**98.3**$^{\pm0.0}$	**98.9**$^{\pm0.0}$	**94.1**$^{\pm0.1}$	**89.6**$^{\pm0.9}$	74.3$^{\pm0.6}$	73.7$^{\pm0.8}$	49.2$^{\pm1.3}$	58.1$^{\pm0.5}$	79.1$^{\pm1.2}$
mBERT	178M	98.4$^{\pm0.0}$	97.3$^{\pm0.1}$	98.3$^{\pm0.0}$	92.2$^{\pm0.1}$	83.5$^{\pm0.6}$	67.9$^{\pm1.2}$	62.7$^{\pm1.2}$	39.6$^{\pm1.3}$	46.4$^{\pm0.7}$	76.5$^{\pm0.9}$
XLM-R$_{\text{base}}$	278M	98.8$^{\pm0.0}$	97.7$^{\pm0.0}$	98.7$^{\pm0.0}$	93.7$^{\pm0.1}$	87.6$^{\pm0.6}$	73.1$^{\pm0.7}$	72.2$^{\pm0.3}$	49.4$^{\pm0.5}$	58.6$^{\pm0.3}$	78.9$^{\pm0.6}$
NorBERT$_{3,\text{ large}}$	353M	**99.1**$^{\pm0.0}$	**98.5**$^{\pm0.0}$	**99.1**$^{\pm0.0}$	**94.6**$^{\pm0.1}$	**91.4**$^{\pm0.5}$	**79.2**$^{\pm0.7}$	**78.4**$^{\pm0.6}$	**54.1**$^{\pm0.6}$	**61.0**$^{\pm0.4}$	**88.7**$^{\pm0.8}$
NB-BERT$_{\text{large}}$	355M	98.7$^{\pm0.0}$	98.2$^{\pm0.1}$	98.3$^{\pm0.1}$	**94.6**$^{\pm0.1}$	89.8$^{\pm0.6}$	**79.2**$^{\pm0.9}$	77.5$^{\pm0.7}$	**54.6**$^{\pm0.7}$	59.7$^{\pm0.1}$	87.0$^{\pm0.5}$
XLM-R$_{\text{large}}$	560M	98.9$^{\pm0.0}$	98.0$^{\pm0.0}$	98.8$^{\pm0.1}$	94.3$^{\pm0.1}$	87.5$^{\pm1.0}$	76.8$^{\pm0.6}$	75.4$^{\pm1.3}$	52.3$^{\pm0.6}$	58.6$^{\pm0.3}$	84.8$^{\pm0.5}$

Table 3: NorBench scores for the existing language models and our novel NorBERT$_3$ family of models. We report the mean and standard deviation statistics over 5 runs; the best results are printed in boldface. The 'Size' column reports the number of parameters in the model; the models are sorted by this value and divided into four size categories. The best results (within one standard deviation) in each category are typeset in bold.

Model	Size	Doc. SA	Sent. SA	NoCoLA	NB-NN
NorT5$_{\text{x-small}}$	32M	**70.1**$^{\pm1.1}$	**55.2**$^{\pm13.6}$	**51.4**$^{\pm0.4}$	**82.1**$^{\pm0.2}$
NorT5$_{\text{small}}$	88M	**73.7**$^{\pm1.4}$	**73.2**$^{\pm0.7}$	**54.4**$^{\pm0.3}$	**85.1**$^{\pm0.1}$
mT5$_{\text{small}}$	300M	24.8$^{\pm3.0}$	22.4$^{\pm0.0}$	25.4$^{\pm5.4}$	33.2$^{\pm0.3}$
North-T5$_{\text{small}}$	300M	20.9$^{\pm0.1}$	22.4$^{\pm0.0}$	33.8$^{\pm7.9}$	36.0$^{\pm0.1}$
T5$_{\text{base}}$	223M	47.2$^{\pm3.5}$	41.3$^{\pm3.2}$	17.6$^{\pm0.8}$	8.9$^{\pm0.0}$
NorT5$_{\text{base}}$	228M	**77.4**$^{\pm0.4}$	**73.4**$^{\pm0.8}$	**58.9**$^{\pm0.3}$	**86.6**$^{\pm0.1}$
mT5$_{\text{base}}$	582M	21.0$^{\pm0.1}$	24.8$^{\pm4.9}$	25.3$^{\pm10.1}$	38.6$^{\pm0.1}$
North-T5$_{\text{base}}$	582M	21.2$^{\pm0.3}$	22.5$^{\pm0.2}$	41.1$^{\pm9.6}$	39.8$^{\pm0.2}$
NorT5$_{\text{large}}$	808M	**77.7**$^{\pm0.5}$	**76.9**$^{\pm2.0}$	**59.4**$^{\pm0.5}$	**86.8**$^{\pm0.1}$
mT5$_{\text{large}}$	1 230M	59.9$^{\pm20.1}$	29.1$^{\pm6.6}$	50.4$^{\pm4.0}$	40.0$^{\pm0.1}$
North-T5$_{\text{large}}$	1 230M	72.9$^{\pm1.2}$	22.4$^{\pm0.0}$	46.8$^{\pm18.7}$	41.1$^{\pm0.1}$

Table 4: NorBench scores for encoder-decoder models, evaluated in a generative text-to-text setting. The best results (within one standard deviation) in each category are typeset in bold.

NorBERT$_{3,\text{ large}}$ (with 353M parameters) specifically obtains the highest scores for most of the tasks except targeted sentiment analysis. At the same time, we see that the smaller models are still very competitive – perhaps most notably NorBERT$_{3,\text{ small}}$ (with 40M parameters) – and there is certainly an aspect of diminishing returns with increasing the number of parameters.

Encoder-decoder scores Table 4 shows the results of T5 models evaluated on four generative tasks. We can see that the performance monotonously improves with scale but the differences between models of different sizes are not drastic. Unfortunately, we found the mT5-based models to be highly unstable and unable to reach decent performance. Our NorT5-large model turned out to be the best across all the tasks.

Gender-bias evaluation Table 5 shows the normative and descriptive occupational bias scores for each model. All models have higher descriptive scores compared to the normative ones, which comes as no surprise. Descriptive scores show how well the models align with the real world distribution of occupations between genders. While no model achieves a perfect score, the top three best models are the NorBERT$_{3,\text{ base}}$ trained on respectively Wikipedia, NAK, and NCC. The nature of these corpora leads to increased correlations between gendered-nouns and occupations, as they usually tend to be described in a descriptive way. NorT5$_{\text{x-small}}$ achieves the worst descriptive bias score of all models, but still scoring better than the best model on the normative score. Looking more specifically at gender-dominated and gender-neutral occupations, it is clear that all models are much better at identifying female-dominated occupations. All models achieve very low scores on gender-neutral occupations, suggesting a tendency to correlate occupations with one gender, rather then equally representing them. These results can be seen in Table 8 in the Appendix.

On the other hand, when we expect genders to be equally represented, no model achieves as high scores in the normative scores, as in the descriptive ones. The best Norwegian model (second best overall) is the smallest model NorBERT$_{3,\text{ x-small}}$,

Model	Normative	Descriptive
NorBERT$_{3,\text{ x-small}}$	19.78	37.36
NorBERT$_{3,\text{ small}}$	8.54	34.92
NorBERT$_1$	16.23	39.31
NorBERT$_2$	3.17	34.67
NB-BERT$_{\text{base}}$	18.55	36.50
ScandiBERT	14.04	43.95
mBERT	**24.66**	41.88
XLM-R$_{\text{base}}$	16.60	36.99
NorBERT$_{3,\text{ base}}$	13.55	39.43
XLM-R$_{\text{large}}$	19.16	46.64
NB-BERT$_{\text{large}}$	11.35	40.90
NorBERT$_{3,\text{ large}}$	13.67	42.73
NorBERT$_{3,\text{ base}}$, oversampled	9.64	36.99
NorBERT$_{3,\text{ base}}$, NAK only	14.04	49.81
NorBERT$_{3,\text{ base}}$, NCC only	12.57	48.84
NorBERT$_{3,\text{ base}}$, mC4 only	11.72	39.31
NorBERT$_{3,\text{ base}}$, NB only	12.33	38.21
NorBERT$_{3,\text{ base}}$, Wiki only	15.99	**50.42**
NorT5$_{\text{x-small}}$	8.91	<u>33.69</u>
NorT5$_{\text{small}}$	<u>0.12</u>	34.06
NorT5$_{\text{base}}$	5.25	43.83
NorT5$_{\text{large}}$	2.56	34.18

Table 5: Normative and descriptive occupational bias scores (Touileb et al., 2023). Best scores are typeset in bold, and worst scores are underlined.

which might suggest that from a normative perspective, the smaller the model, the more balanced representation of genders, at least when it comes to occupations. The best scoring model is the multilingual mBERT model. On a closer analysis, it is apparent that mBERT is very good at correlating occupations with the male gender (similarly to the descriptive score in Table 8 in the Appendix), which seems to skew the metric. This might exhibit a weakness in the metric, where models skewed towards one gender can get higher overall scores even if they fail to represent the other gender.

Harmfulness scores In addition to the normative and descriptive occupational bias scores, we also compute the harmfulness of the sentence-completions generated by these models. Table 6 shows the HONEST scores (Nozza et al., 2021) of each model. Here we evaluate the top-k completions, where we look at the first, five, ten, and twenty most likely completions. Overall, NorBERT$_3$ and NorT5 models achieve very low harmfulness scores compared to the other Norwegian language models. All NorT5 models do not return harmful words as the most likely completions, and are overall generating few problematic

Model	k = 1	k = 5	k = 10	k = 20
NorBERT$_{3,\text{ x-small}}$	0.0062	0.0062	0.0040	0.0037
NorBERT$_{3,\text{ small}}$	0.0015	0.0018	0.0027	0.0049
NorBERT$_1$	0.0310	<u>0.0378</u>	<u>0.0306</u>	<u>0.0258</u>
NorBERT$_2$	0.0356	0.0229	0.0189	0.0159
NB-BERT$_{\text{base}}$	0.0124	0.0083	0.0080	0.0069
ScandiBERT	**0.0**	0.0010	0.0043	0.0045
mBERT	**0.0**	0.0028	0.0057	0.0068
XLM-R$_{\text{base}}$	<u>0.0450</u>	0.0169	0.0117	0.0128
NorBERT$_{3,\text{ base}}$	**0.0**	0.0027	0.0026	0.0055
XLM-R$_{\text{large}}$	0.0342	0.0158	0.0131	0.0116
NB-BERT$_{\text{large}}$	0.0294	0.0285	0.0279	0.0244
NorBERT$_{3,\text{ large}}$	**0.0**	0.0006	0.0013	0.0033
NorBERT$_{3,\text{ base}}$, oversampled	0.0046	0.0071	0.0085	0.0092
NorBERT$_{3,\text{ base}}$, NAK only	0.0093	0.0080	0.0093	0.0125
NorBERT$_{3,\text{ base}}$, NCC only	**0.0**	0.0006	0.0010	0.0028
NorBERT$_{3,\text{ base}}$, mC4 only	**0.0**	0.0003	**0.0009**	0.0038
NorBERT$_{3,\text{ base}}$, NB only	0.0015	0.0031	0.0012	**0.0026**
NorBERT$_{3,\text{ base}}$, Wiki only	**0.0**	0.0012	0.0071	0.0082
NorT5$_{\text{x-small}}$	**0.0**	0.0010	0.0018	**0.0026**
NorT5$_{\text{small}}$	**0.0**	0.0003	0.0018	0.0037
NorT5$_{\text{base}}$	**0.0**	0.0010	0.0077	0.0090
NorT5$_{\text{large}}$	**0.0**	**0.0**	0.0014	0.0037

Table 6: The harmfulness score of models looking at top one, five, ten, and twenty most likely completions using HONEST (Nozza et al., 2021). The best scores are in bold, while the worst are underlined.

completions. However, since the HONEST score relies on lexicons, some completions not included in these might still be harmful. XLM-R$_{\text{base}}$ is the worst model in top one completions, while the NorBERT$_1$ is the worst model across all remaining top k completions.

5.2 Comparison of Norwegian corpora

The downstream performance of a language model is a result of a combination of training choices and choices of the training corpus. In order to study the second aspect, we fix the training configuration and pre-train multiple NorBERT$_{3,\text{ base}}$ models on different Norwegian corpora.

We compare a simple concatenation of all available resources (*'combined'*) against a variant with oversampling the quality data. The reasoning behind this was that the mC4 corpus is the most noisy of all the above, since it is created by web crawling. We hypothesized that artificially increasing the amount of data from the cleaner corpora (Wikipedia, NBDigital, NCC and NAK) will improve the resulting model's performance. We implemented this by creating an *'oversampled'* train collection where all the sentences from the clean corpora were repeated twice, so that the total size of the 'clean' part approximately matched the size of

Corpus	UPOS	UFeats	Lemma	LAS	NER	Doc. SA	Sent. SA	TSA	NoCoLA	NorQuAD
Combined	**99.0**$^{\pm 0.0}$	**98.3**$^{\pm 0.1}$	**98.8**$^{\pm 0.0}$	**94.2**$^{\pm 0.1}$	89.4$^{\pm 0.7}$	76.2$^{\pm 0.8}$	74.4$^{\pm 0.3}$	**52.2**$^{\pm 0.7}$	**59.2**$^{\pm 0.3}$	**86.2**$^{\pm 0.3}$
Oversampled	98.9$^{\pm 0.0}$	**98.2**$^{\pm 0.0}$	98.7$^{\pm 0.0}$	94.1$^{\pm 0.1}$	**90.5**$^{\pm 0.3}$	75.0$^{\pm 0.4}$	75.2$^{\pm 0.5}$	50.4$^{\pm 0.4}$	57.6$^{\pm 0.1}$	83.4$^{\pm 0.7}$
NAK	98.9$^{\pm 0.0}$	98.0$^{\pm 0.0}$	98.5$^{\pm 0.0}$	94.1$^{\pm 0.1}$	90.4$^{\pm 0.6}$	**76.9**$^{\pm 0.1}$	**77.5**$^{\pm 0.9}$	51.3$^{\pm 0.7}$	58.3$^{\pm 0.3}$	82.5$^{\pm 0.4}$
NCC	**99.0**$^{\pm 0.0}$	**98.2**$^{\pm 0.0}$	98.7$^{\pm 0.0}$	**94.3**$^{\pm 0.1}$	89.5$^{\pm 0.6}$	74.8$^{\pm 0.3}$	74.8$^{\pm 1.4}$	50.0$^{\pm 0.5}$	58.3$^{\pm 0.4}$	83.0$^{\pm 1.2}$
mC4	**99.0**$^{\pm 0.0}$	98.1$^{\pm 0.0}$	98.7$^{\pm 0.0}$	**94.2**$^{\pm 0.1}$	**90.2**$^{\pm 0.5}$	76.3$^{\pm 0.6}$	76.8$^{\pm 0.7}$	50.8$^{\pm 0.9}$	58.5$^{\pm 0.3}$	83.2$^{\pm 0.5}$
Wiki	98.9$^{\pm 0.0}$	97.6$^{\pm 0.0}$	98.3$^{\pm 0.0}$	93.6$^{\pm 0.1}$	87.9$^{\pm 0.3}$	71.9$^{\pm 1.0}$	68.9$^{\pm 1.2}$	44.9$^{\pm 0.4}$	54.1$^{\pm 0.3}$	78.2$^{\pm 0.5}$
NBDigital	98.9$^{\pm 0.0}$	98.0$^{\pm 0.0}$	98.7$^{\pm 0.0}$	93.9$^{\pm 0.1}$	87.1$^{\pm 0.7}$	72.7$^{\pm 0.4}$	70.1$^{\pm 0.5}$	45.2$^{\pm 0.9}$	56.1$^{\pm 0.1}$	79.3$^{\pm 0.6}$

Table 7: The downstream performance of NorBERT$_{3,\text{ base}}$ models pre-trained on different corpora. We report the mean and standard deviation statistics over 5 runs; the best results (within one standard deviation) are shown in boldface.

the mC4 corpus. In addition, to study the respective usefulness of particular Norwegian text collections, we trained separate models from scratch on NAK, NCC, mC4, Wikipedia, and NBDigital.

Corpora comparison results Table 7 shows the results. We believe there are two noteworthy – and perhaps surprising – take-aways:

1. We hypothesised that oversampling the high-quality texts should lead to increased downstream performance. This is evidently a false assumption as oversampling works slightly worse overall. Large language models are known to be sensitive to duplicate data (Lee et al., 2022), which might explain such a behavior.
2. A straightforward concatenation of all available resources does not necessarily lead to better performance – but it is a reasonable approach for a general model as it works the best on average. On the other hand, pre-training only on NAK leads to substantially improved performance on sentiment analysis, perhaps due to a closer match in terms of text type.

6 Related work

Evaluating pre-trained language models for particular languages and cross-lingualy is a venerable research sub-field within NLP. Well-known benchmark sets for English include GLUE (Wang et al., 2018), SuperGLUE (Wang et al., 2019), and GLGE (Liu et al., 2021), among others. However, up to now benchmarking LMs for Norwegian was limited to separate test sets with non-standardised evaluation workflows. ScandEval (Nielsen, 2023) aims to create a standard natural language understanding benchmark across Scandinavian languages (Danish, Swedish, and Norwegian). However, it does not focus on evaluating specifically Norwegian tasks and half of its Norwegian benchmarks (linguistic acceptability and question answering) are not human-annotated. We address this issue with NorBench.

7 Future work

We consider NorBench to be a dynamic resource that we plan to continually extend in future work, to support additional tasks and additional architectures. While we anticipate including new annotated benchmark data as they may become available in the future, there are also existing datasets that we plan to include in the shorter term, like coreference resolution based on the NARC dataset (Mæhlum et al., 2022) and negation resolution based on NoReC$_{neg}$ (Mæhlum et al., 2021). Finally, we also plan on adding tasks that more specifically target generative models, including sequence-generation tasks like summarization, but also prompt-based formulations of the existing NorBench tasks for few-shot evaluation.

8 Summary

In this paper we have presented NorBench, a set of standardized benchmark tasks for systematically evaluating and comparing Norwegian language models. The aim of this effort is to provide NLP practitioners with a comprehensive and streamlined service, including a leaderboard, human-annotated datasets, evaluation workflow, and open code implementing this workflow.

This paper also describes and evaluates a set of novel NorBERT$_3$ masked LMs trained on several different Norwegian text collections in different model sizes. They are shown to outperform Norwegian LMs from prior work on the majority of NorBench tasks.

Acknowledgements

NorBench forms part of NorLM, an initiative coordinated by the Language Technology Group (LTG) at the University of Oslo (UiO), with the goal of developing resources for large-scale language modeling for Norwegian. The efforts described in the current paper were jointly funded by the SANT project (Sentiment Analysis for Norwegian; coordinated by LTG at UiO and funded by the the Research Council of Norway, grant number 270908), and the HPLT project (High Performance Language Technologies; coordinated by Charles University). The computations were performed on resources provided through Sigma2 – the national research infrastructure provider for High-Performance Computing and large-scale data storage in Norway.

Parts of this work was supported by industry partners and the Research Council of Norway with funding to MediaFutures: Research Centre for Responsible Media Technology and Innovation, through the Centres for Research-based Innovation scheme, project number 309339.

References

Elisa Bassignana, Valerio Basile, and Viviana Patti. 2018. Hurtlex: A multilingual lexicon of words to hurt. In *Proceedings of the 5th Italian Conference on Computational Linguistics, CLiC-it 2018*, volume 2253, pages 1–6. CEUR-WS.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep bi-affine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Sardana Ivanova, Fredrik Aas Andreassen, Matias Jentoft, Sondre Wold, and Lilja Øvrelid. 2023. NorquAD: Norwegian question answering dataset. In *The 24th Nordic Conference on Computational Linguistics*.

Matias Jentoft and David Samuel. 2023. NocoLA: The norwegian corpus of linguistic acceptability. In *The 24rd Nordic Conference on Computational Linguistics*.

Fredrik Jørgensen, Tobias Aasmoe, Anne-Stine Ruud Husevåg, Lilja Øvrelid, and Erik Velldal. 2020. NorNE: Annotating Named Entities for Norwegian. In *Proceedings of the 12th Edition of the Language Resources and Evaluation Conference*, Marseille, France, 2020.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Per E Kummervold, Javier De la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. 2021. Operationalizing a national digital library: The case for a Norwegian transformer model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 20–29, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-scale contextualised language modelling for Norwegian. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 30–40, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training

data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain.

Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, Pengcheng Wang, Jiusheng Chen, Daxin Jiang, Jiancheng Lv, Ruofei Zhang, Winnie Wu, Ming Zhou, and Nan Duan. 2021. GLGE: A new general language generation evaluation benchmark. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 408–420, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Petter Mæhlum, Jeremy Barnes, Robin Kurtz, Lilja Øvrelid, and Erik Velldal. 2021. Negation in Norwegian: an annotated dataset. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics*.

Petter Mæhlum, Dag Haug, Tollef Jørgensen, Andre Kåsen, Anders Nøklestad, Egil Rønningstad, Per Erik Solberg, Erik Velldal, and Lilja Øvrelid. 2022. Narc–norwegian anaphora resolution corpus. In *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 48–60.

B.W. Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451.

Dan Saattrup Nielsen. 2023. Scandeval: A benchmark for Scandinavian natural language processing. In *The 24rd Nordic Conference on Computational Linguistics*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2021. HONEST: Measuring hurtful sentence completion in language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2398–2406, Online. Association for Computational Linguistics.

Lilja Øvrelid and Petter Hohle. 2016. Universal Dependencies for Norwegian. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, pages 1579–1585, Portorož, Slovenia.

Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2020. A fine-grained sentiment dataset for Norwegian. In *Proceedings of the 12th Edition of the Language Resources and Evaluation Conference*, Marseille, France.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William S. Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling Language Models: Methods, Analysis & Insights from Training Gopher. *CoRR*, abs/2112.11446.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text

transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Egil Rønningstad, Erik Velldal, and Lilja Øvrelid. 2022. Entity-level sentiment analysis (ELSA): An exploratory task survey. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6773–6783, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: Bert meets british national corpus.

Sam Shleifer and Myle Ott. 2022. Normformer: Improved transformer pretraining with extra normalization.

Per Erik Solberg, Arne Skjærholt, Lilja Øvrelid, Kristin Hagen, and Janne Bondi Johannessen. 2014. The Norwegian Dependency Treebank. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, Reykjavik, Iceland.

Kari Tenfjord, Paul Meurer, and Knut Hofland. 2006. The ASK corpus - a language learner corpus of Norwegian as a second language. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Samia Touileb and Debora Nozza. 2022. Measuring harmful representations in Scandinavian language models. In *Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS)*, pages 118–125, Abu Dhabi, UAE. Association for Computational Linguistics.

Samia Touileb, Lilja Øvrelid, and Erik Velldal. 2022. Occupational biases in Norwegian and multilingual language models. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 200–211, Seattle, Washington. Association for Computational Linguistics.

Samia Touileb, Lilja Øvrelid, and Erik Velldal. 2023. Measuring normative and descriptive biases in language models using census data.

Erik Velldal, Lilja Øvrelid, Eivind Alexander Bergem, Cathrine Stadsnes, Samia Touileb, and Fredrik Jørgensen. 2018. NoReC: The Norwegian Review Corpus. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference*, pages 4186–4191, Miyazaki, Japan.

Erik Velldal, Lilja Øvrelid, and Petter Hohle. 2017. Joint UD parsing of Norwegian Bokmål and Nynorsk. In *Proceedings of the 21st Nordic Conference of Computational Linguistics*, pages 1–10, Gothenburg, Sweden.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

A Sentiment analysis classification – details

As mentioned above, initially reviews were rated on a scale from 1 to 6 but later we have narrowed down the classes to 3 (classes 1, 2, and 3 mapped to the 'negative' class, class 4 to the 'fair' class, and classes 5 and 6 to the 'positive' class).

Document-level sentiment analysis is complicated by 2 factors. First, 40% of all the dataset texts were longer than 512 (white-space separated) tokens with the maximum text length reaching 3 943 tokens. Second, the average text length as well as the number of samples in NoReC increased from negative to positive classes. Therefore we had a challenging task, with the 'negative' class having shorter texts and a smaller sample size, compared to the 'fair' class with larger texts and more samples, as well as 'positive' class with the most of everything. Several feature engineering strategies were attempted for baseline document-level sentiment analysis, but the most straightforward approach proved the most effective: to simply truncate all texts to the first 512 sub-words. Such a truncation is used for all sequence classification tasks.

B Descriptive bias scores

Model	N	F	M
$\text{NorBERT}_{3,\text{ x-small}}$	2.19	31.01	4.15
$\text{NorBERT}_{3,\text{ small}}$	0.48	33.69	0.73
$\text{NorBERT}_{3,\text{ base}}$	1.22	33.21	5.00
$\text{NorBERT}_{3,\text{ large}}$	1.70	33.33	7.69
mBERT	**3.41**	6.47	**31.99**
ScandiBERT	0.97	16.23	26.73
$\text{XLM-R}_{\text{base}}$	1.70	23.32	11.96
$\text{XLM-R}_{\text{large}}$	2.07	18.07	26.49
$\text{NorBERT}_{3,\text{ base}}$, oversampled	0.36	33.45	3.17
$\text{NorBERT}_{3,\text{ base}}$, NAK only	2.56	28.81	18.43
$\text{NorBERT}_{3,\text{ base}}$, NCC only	2.19	30.76	15.87
$\text{NorBERT}_{3,\text{ base}}$, mC4 only	0.61	33.33	5.37
$\text{NorBERT}_{3,\text{ base}}$, NB only	0.73	30.03	7.44
$\text{NorBERT}_{3,\text{ base}}$, Wiki only	2.56	25.88	21.97
$\text{NorT5}_{\text{x-small}}$	0.48	32.71	0.48
$\text{NorT5}_{\text{small}}$	0.0	**34.06**	0.0
$\text{NorT5}_{\text{base}}$	0.36	16.97	26.49
$\text{NorT5}_{\text{large}}$	0.12	**34.06**	0.0

Table 8: Descriptive bias scores of gender-dominated and gender-neutral occupations. Where N stands for neutral, F for female, and M for male. Best score are typeset in bold, and worst scores are underlined.

C Hyperparameters

Hyperparameter	NorBERT$_{3,\text{ x-small / small / base / large}}$
Number of layers	12 / 12 / 12 / 24
Hidden size	192 / 384 / 768 / 1 024
FF intermediate size	512 / 1 024 / 2 048 / 2 730
Vocabulary size	50 000
Attention heads	3 / 6 / 12 / 16
Dropout	0.1
Attention dropout	0.1
Training steps	250 000
Batch size	8 192
Sequence length	512
Warmup steps	4 000 (1.6% steps)
Initial learning rate	0.01
Final learning rate	0.001
Learning rate decay	cosine
Weight decay	0.1
Layer norm ϵ	1e-7
Optimizer	LAMB
LAMB ϵ	1e-6
LAMB β_1	0.9
LAMB β_2	0.98
Gradient clipping	2.0

Table 9: Pre-training hyperparameters. The models differ only in their hidden size and number of layers, the learning rate schedule and other training settings are kept identical.

Hyperparameter	Value
Dropout	0.1
Attention dropout	0.1
Label smoothing	0.1
Epochs	10
Max length	512
Batch size	32
Warmup steps	250
Initial learning rate	0.001
Final learning rate	0.0001
Learning rate decay	cosine
Weight decay	0.1
Optimizer	AdamW
Gradient clipping	10.0

Table 10: Hyperparameters for fine-tuning language models on UD tasks.

Hyperparameter	Value
Dropout	0.1
Attention dropout	0.1
Epochs	10
Max length	512
Batch size	32
Learning rate	5e-5
Learning rate decay	constant
Weight decay	0.01
Optimizer	AdamW

Table 11: Hyperparameters for fine-tuning language models on NER and TSA.

Hyperparameter	Value
Dropout	0.1
Attention dropout	0.1
Epochs	10
Max length	512
Batch size	16
Initial learning rate	1e-5
Learning rate decay	constant
Weight decay	0.01
Optimizer	AdamW

Table 12: Hyperparameters for fine-tuning language models on document-level and sentence-level sentiment analysis.

Hyperparameter	Value
Dropout	0.1
Attention dropout	0.1
Epochs	10
Max length	512
Batch size	32
Warmup portion	6%
Initial learning rate	1e-5
Final learning rate	1e-6
Learning rate decay	cosine
Weight decay	0.01
Optimizer	AdamW

Table 13: Hyperparameters for fine-tuning language models on NoCoLA.

Hyperparameter	Value
Dropout	0.0
Attention dropout	0.0
Epochs	10
Max length	512
Batch size	32
Warmup portion	6%
Initial learning rate	2e-5
Final learning rate	2e-6
Learning rate decay	cosine
Weight decay	0.1
Optimizer	AdamW

Table 14: Hyperparameters for fine-tuning language models on Bokmål–Nynorsk machine translation.

Hyperparameter	Value
Dropout	0.1
Attention dropout	0.1
Epochs	3
Batch size	16
Warmup steps	100
Max length	384
Document stride	128
Initial learning rate	1e-4
Final learning rate	0.0
Learning rate decay	linear
Weight decay	0.01
Optimizer	AdamW

Table 15: Hyperparameters for fine-tuning language models on NorQuAD

Making Instruction Finetuning Accessible to Non-English Languages: A Case Study on Swedish

Oskar Holmström*
Linköping University
oskar.holmstrom@liu.se

Ehsan Doostmohammadi*
Linköping University
ehsan.doostmohammadi@liu.se

Abstract

In recent years, instruction-finetuned models have received increased attention due to their remarkable zero-shot and generalization capabilities. However, the widespread implementation of these models has been limited to the English language, largely due to the costs and challenges associated with creating instruction datasets. To overcome this, automatic instruction generation has been proposed as a resourceful alternative. We see this as an opportunity for the adoption of instruction finetuning for other languages. In this paper we explore the viability of instruction finetuning for Swedish. We translate a dataset of generated instructions from English to Swedish, using it to finetune both Swedish and non-Swedish models. Results indicate that the use of translated instructions significantly improves the models' zero-shot performance, even on unseen data, while staying competitive with strong baselines ten times in size. We see this paper is a first step and a proof of concept that instruction finetuning for Swedish is within reach, through resourceful means, and that there exist several directions for further improvements.

1 Introduction

The use of pretrained language models in natural language processing (NLP) is widespread, with finetuning or zero-shot approaches employed for various tasks. However, not all pretrained models exhibit strong zero-shot performance or are cost-effective to finetune for every new task. To overcome these limitations, instruction finetuning—finetuning on natural language processing tasks that are described as instructions—has been demonstrated to enhance generalization to unseen NLP problems and tasks (Wei et al., 2022; Chung et al., 2022). Instruction finetuning, although beneficial, can be costly since it requires human annotation or feedback. To overcome this issue, automatic instruction generation has been demonstrated as a cost-effective alternative (Honovich et al., 2022; Wang et al., 2022a). While the benefits of automatic finetuning are substantial for English, which has abundant data resources, they are even more pronounced for languages with limited resources, such as Swedish.

In this paper, we explore how automatic methods for instruction finetuning can be extended to Swedish. The work is partly based on Unnatural Instructions (Honovich et al., 2022), a method of bootstrapping the instruction creation process. We use the generated instruction as a teacher to a Swedish student, where a translator module acts as an intermediary. The dataset is translated from English to Swedish and then used to finetune various Swedish and non-Swedish models to investigate the effectiveness of the proposed technique. The translations and models are evaluated using both human and automatic methods.

We find that the translated instructions generates a significant increase in zero-shot performance, even to unseen data. This paper is a first step, and a proof of concept, that instruction finetuning for Swedish is possible and that there exist several directions for further improvements.

2 Related Work

Language models have demonstrated the capability to solve tasks through following instructions in a zero-shot setting. However, their performance can be enhanced by finetuning on a diverse set of task-specific instruction data. This allows the model to adapt and generalize to new, unseen tasks, reducing the need for task-specific finetuning and enabling

*Equal contribution.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 634–642

an off-the-shelf solution (Weller et al., 2020; Efrat and Levy, 2020; Mishra et al., 2022; Sanh et al., 2022; Chakrabarty et al., 2022; Gupta et al., 2022; Wang et al., 2022b). Manually procuring data for task specific finetuning can be costly. To mitigate this issue, researchers have explored automatically generating data (Schick and Schütze, 2021). Studies have shown that this alternative is highly effective, with performance that is only slightly behind that of large language models, which has been finetuned on manual data (Honovich et al., 2022; Wang et al., 2022a).

3 Automatic Instruction Finetuning for Swedish

We use two different instruction specific datasets and translate them to Swedish: The first one to finetune two models, and the second one as a held-out evaluation set. We evaluate the performance of the two models before and after instruction finetuning together with a strong GPT3 baseline to explore the usability of the automatically procured and translated instruction data.

The code, model checkpoints, and datasets used in the paper are made available[1].

3.1 Datasets

UNNATURAL INSTRUCTIONS In our study, we use the core dataset of UNNATURAL INSTRUCTIONS (Honovich et al., 2022) as a base for training and testing the models. The dataset was generated by starting with 15 manually written samples as seed, and then incrementally adding more samples with OpenAI's `text-DaVinci-002`, using three seed examples to generate a fourth one at a time. Each sample contains the following parts: (1) the instruction, which is the definition of the task, e.g., "Find an answer to the mathematical problem."; (2) the input text which is a specific example in the instruction space, e.g., "A wheel has a circumference of 15 feet. What is its diameter in inches?"; (3) constraints, which specifies the restrictions of the expected answer, e.g., "The output should be a number, rounded off to 2 decimal places."; and (4) the output, which is the correct generation considering all the previous instructions and constraints.

The core set of the UNNATURAL INSTRUCTIONS dataset comprises 68,478 samples, which we split into two sets: 100 samples for testing and the remainder for training. The top 10 tasks in the dataset belong to a broad set of categories, and are as follows: question answering, sentiment analysis, arithmetic, geometry, event ordering, fact verification, fill-in-the-blank, general math puzzles, identifying overlapping strings, and array manipulations and puzzles.

NATURAL INSTRUCTIONS For evaluation of our models, we utilize a subset of the NATURAL INSTRUCTIONS dataset generated by human annotators (Mishra et al., 2022). The test set of this dataset comprises 12 tasks, and we randomly select 80 samples from each task to assess the models' performance using ROUGE-L, and a subset of randomly selected 5 sample per task for human evaluation. The tasks are question and answer generation with regards to different aspects of an incident. For example, "Jack played basketball after school, after which he was very tired. Question: How long did Jack play basketball?". The task descriptions and the expected generated answers are also longer on average, resulting in a more difficult test set compared to UNNATURAL INSTRUCTIONS. See Appendix A for an overview of the tasks.

Automatic Translation For the automatic translation of the data into Swedish, we use off-the-shelf machine translation models. The UNNATURAL INSTRUCTIONS dataset is translated with DeepL[2] and the NATURAL INSTRUCTIONS dataset is translated with `GPT3-DaVinci-003`. To assess the quality of the translations, we conduct a human evaluation, which rates the translations from one to three based on two criteria: (1) grammaticality and naturalness, and (2) accuracy compared to the source text. 120 random samples were selected from the UNNATURAL INSTRUCTIONS dataset and 10 examples per task were selected from the NATURAL INSTRUCTIONS dataset. The evaluator rates the translations on a scale of 1 to 3, with 1 indicating significant errors, 2 indicating minor errors, and 3 indicating correct and natural translations. The results show an average rating of 2.75 for grammaticality and naturalness, and 2.41 for accuracy for the UNNATURAL INSTRUCTIONS dataset, and 2.83 and 2.55 for the NATURAL INSTRUCTIONS dataset.

Perplexity Dataset In order to evaluate the quality of the models outlined in Section 3.2, we assess their perplexity. To guarantee that the generated

[1] https://github.com/oskarholmstrom/sweinstruct

[2] https://www.deepl.com

Model	Perplexity
GPT-SW3	1.92
GPT-SW3-UI	2.66
OPT	2.79
OPT-UI	5.45
GPT3-Curie	2.41
GPT3-Curie-I	2.99
GPT3-DaVinci	1.91
GPT3-DaVinci-I	1.94

Table 1: Perplexity of all the models on the SVT dataset.

Model	UI ROUGE-L	NI ROUGE-L
GPT-SW3	0.084	0.009
GPT-SW3-UI	**0.542**	0.124
OPT	0.071	0.006
OPT-UI	0.449	0.101
GPT3-Curie	0.060	0.030
GPT3-Curie-I	0.308	0.108
GPT3-DaVinci	0.083	0.026
GPT3-DaVinci-I	0.537	**0.151**

Table 2: ROUGE-L scores on UNNATURAL INSTRUCTIONS (UI) and NATURAL INSTRUCTIONS (NI) test sets for all the models. The best results are in bold.

texts are of high quality, and to eliminate the possibility of evaluating the models on data that was part of their pretraining, we use a custom dataset comprised of current news articles from the Swedish national public television broadcaster, SVT[3]. Our dataset is made up of 357 articles covering a range of subjects, with an average length of 256 tokens per article. These articles were published between July 1st, 2022, and January 19th, 2023.

3.2 Models

GPT-SW3 (Ekgren et al., 2022) is a GPT2-like (Radford et al.) model pretrained on the Nordic Pile, where 26% of the data is Swedish (**?**). The model is available in different sizes, but as a proof of concept, we finetune and evaluate the model with 1.3B parameters.

OPT (Zhang et al., 2022) is a BartDecoder-like (Lewis et al., 2020) language model pretrained predominantly on English data. However, even models that are intended to be trained on English are exposed to other languages during pretraining due to language contamination (Blevins and Zettlemoyer, 2022). We choose to finetune OPT to gain a perspective on how the predominant language in the base model affects its instruction handling abilities. To allow for fair comparisons, we use the 1.3B parameter model. The model is openly available and is trained on publicly available data.

GPT3 (Brown et al., 2020) is a proprietary, closed-source large language model. We use both the pre-trained and instruction tuned GPT3-DaVinci-003, which has 175B parameters, and GPT3-Curie-001, which has 6.7B parameters. We abbreviate the instruction finetuned versions with "-I".

[3]https://www.svt.se

3.3 Finetuning and Experimental Setup

Having the training and test data described in Section 3.1, we instruction finetune the GPT-SW3 and the OPT models described in Section 3.2. We call the new models GPT-SW3-UI and OPT-UI, as they are finetuned on the UNNATURAL INSTRUCTIONS (UI) dataset. The models are finetuned using a next token prediction objective for the output, given the description, input, and the constraints of the task. We do not calculate any loss on the three first parts of the sample and the output is attention-masked so that the models cannot gain information from the output. We finetune the models for 3 epochs, following (Honovich et al., 2022). As for the other hyperparameters of the model, we chose 2e-5 for learning rate, 0.1 for weight decay, and 0.1 for warm-up ratio with an AdamW optimizer (Loshchilov and Hutter, 2019). For generation during inference, we use beam search with beam size 4 and 0.75 for temperature.

4 Results

Perplexity We first start with a perplexity analysis to measure the language modelling quality of the models on unseen Swedish data using the dataset described in Section 3.1. When evaluating perplexity using token length normalization, tokenizers that generate sentences with more tokens are favored. However, this approach can be problematic in cross-lingual settings, where tokenizing unknown words may increase the number of tokens generated. To overcome this issue, we use character length normalization as it provides a fairer measurement of perplexity across languages (Liang et al., 2022; Yong et al., 2022). From the results in Table 1, we see that perplexity increases after in-

Model	Natural	Related	Correct
GPT-SW3-UI	2.39	2.38	1.84
OPT-UI	2.25	1.59	1.33
GPT3-Curie-I	2.45	2.38	1.76
GPT3-DaVinci-I	2.56	2.68	2.28

Table 3: Average score of generations from human evaluation of the models on the Natural Instructions dataset.

struction finetuning. It is especially pronounced for the smaller models, while the change is minimal for GPT3-DaVinci. OPT, not originally trained on Swedish, is capable of modelling Swedish but also seems to suffer the most from instruction finetuning.

ROUGE-L Following Wang et al. (2022a), Mishra et al. (2022), and Honovich et al. (2022), we use ROUGE-L to automatically measure the performance of the models. The results show that the non-instruction-finetuned models perform the worst for both datasets. With instruction finetuning, we observe an increase in the ROUGE-L scores, even a major increase for non-Swedish models. It is important to note that the GPT3 models have *not* undergone instruction finetuning on our data. The results highlight the challenge of NATURAL INSTRUCTION tasks compared to UNNATURAL INSTRUCTIONS, partly due to task complexity and partially due to the length of the answers. For a breakdown of the results on all the tasks, refer to Table 4.

Human Evaluation We perform a human evaluation study to accompany the ROUGE-L score analysis of model generations quality. The evaluation was done on 5 randomly selected generations. Three annotators independently scored (1 = not true, 2 = somewhat true, 3 = true) the generations on three different criteria: (1) whether it is natural and grammatically correct; (2) whether it relates to the provided context; (3) if it is a correct answer for the given task. The instruction finetuned DaVinci model with 175B parameters produces more natural and correct outputs than the other models, while GPT-SW3 and the instruction finetuned Curie model perform close to each other on all three criteria. The results are shown in Table 3.

5 Discussion

The results from the perplexity analysis show an increase for all models after instruction finetuning, even though the models become more capable at a broad set of tasks. It has been shown that perplexity does not correlate strongly with downstream task or prompting performance (Liang et al., 2022; Yong et al., 2022). However, when using automatically translated data, we need to be aware that noise in the translation process can affect the models' capabilities. The increase in perplexity could partly be explained by unwanted noise. A hypothesis for why we see a larger increase in perplexity for the OPT model than the GPT-SW3 model is that stronger foundations in the target language makes the model more robust to translation errors.

The significant increase of ROUGE-L scores for all models, especially on the difficult NATURAL INSTRUCTIONS tasks, show that the models can become strong zero-shot generalizers with relatively little finetuning. Unsurprisingly, the stronger performance of GPT-SW3 than OPT shows that a strong foundation in the target language is helpful.

There are some issues with making direct comparisons with the baseline GPT3 models: we do not know what data it has seen during training, the models have not been trained specifically for Swedish, and they have followed a more structured instruction-finetuning process. What can be said is that translated automatic instruction seems to be highly useful. GPT-SW3 outperforms the larger Curie model on both datasets and even the DaVinci model, two orders of magnitude larger, on the unnatural instructions test set. However, our human evaluation shows that there are still significant improvements that need to be made to reach parity with the largest model.

6 Conclusion and Future Work

Using automatically created instructions that have been translated to Swedish provides a significant increase in zero-shot performance when instruction finetuning a GPT-SW3 and OPT model. The GPT-SW3 model shows competitive performance against the hundred times larger instruction-tuned GPT3-DaVinci model. While the results are promising, this is still a work in progress. Human evaluations show that there is significant progress to be made, especially in giving correct answers to instructions. A possible path for performance gain is to study the effects of translation quality

on models' performance. We also leave for future work how the automatically translated instruction finetuning interacts with increased model scale.

In conclusion, we find that instruction finetuning for Swedish is not only within our reach, but it can be achieved with a completely automatic process that yields significant improvements on a broad set of tasks in a zero-shot setting.

Acknowledgments

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by the Berzelius resources provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Center.

References

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explain the cross-lingual capabilities of english pretrained models.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tuhin Chakrabarty, Vishakh Padmakumar, and He He. 2022. Help me write a poem: Instruction tuning as a vehicle for collaborative poetry writing. *arXiv preprint arXiv:2210.13669*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Avia Efrat and Omer Levy. 2020. The turking test: Can language models understand instructions? *arXiv preprint arXiv:2010.11982*.

Ariel Ekgren, Amaru Cuba Gyllensten, Evangelia Gogoulou, Alice Heiman, Severine Verlinden, Joey Öhman, Fredrik Carlsson, and Magnus Sahlgren. 2022. Lessons learned from gpt-sw3: Building the first large-scale generative language model for swedish. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3509–3518.

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey P Bigham. 2022. Improving zero and few-shot generalization in dialogue through instruction tuning. *arXiv preprint arXiv:2205.12673*.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022a. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022b. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *URL https://arxiv. org/abs/2204.07705*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Orion Weller, Nicholas Lourie, Matt Gardner, and Matthew E. Peters. 2020. Learning from task descriptions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1361–1375, Online. Association for Computational Linguistics.

Zheng-Xin Yong, Hailey Schoelkopf, Niklas Muennighoff, Alham Fikri Aji, David Ifeoluwa Adelani, Khalid Almubarak, M Saiful Bari, Lintang Sutawika, Jungo Kasai, Ahmed Baruwa, et al. 2022. Bloom+ 1: Adding language support to bloom for zero-shot prompting. *arXiv preprint arXiv:2212.09535*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Model	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
GPT-SW3	0.035	0.006	0.002	0.013	0.008	0.007	0.000	0.001	0.004	0.021	0.008	0.002
GPT-SW3-UI	0.033	0.204	0.085	0.106	0.067	0.156	0.286	0.080	0.125	0.083	0.080	0.187
OPT	0.060	0.002	0.004	0.0	0.003	0.001	0.001	0.003	0.000	0.000	0.001	0.000
OPT-UI	0.038	0.125	0.052	0.038	0.033	0.044	0.352	0.093	0.187	0.054	0.075	0.120
GPT-Curie	0.027	0.039	0.009	0.030	0.028	0.036	0.034	0.043	0.007	0.040	0.036	0.037
GPT-Curie-I	0.080	0.095	0.067	0.069	0.086	0.106	0.116	0.226	0.217	0.109	0.073	0.059
GPT-DaVinci	0.027	0.038	0.012	0.017	0.028	0.044	0.012	0.020	0.010	0.038	0.032	0.037
GPT-DaVinci-I	0.119	0.071	0.084	0.238	0.131	0.091	0.147	0.419	0.264	0.121	0.093	0.037

Table 4: A breakdown of ROUGE-L scores on the NATURAL INSTRUCTIONS (NI) test subsets for all the models.

A NATURAL INSTRUCTIONS Tasks

The following are a summary of the task descriptions in the NATURAL INSTRUCTIONS test set. Please refer to Mishra et al. (2022) for more information.

1. **Task 1**: Writing questions that require tracking entity references.
2. **Task 2**: Writing answers to questions involving multiple references to the same entity.
3. **Task 3**: Writing questions that ask about event duration, based on given sentences.
4. **Task 4**: Writing answers to questions that involve event duration, based on given sentences.
5. **Task 5**: Writing *implausible* answers to questions that involve event duration, based on given sentences.
6. **Task 6**: Writing questions that involve "transient vs. stationary" events[4].
7. **Task 7**: Writing answers to a questions that involve events that are stationary or transient.
8. **Task 8**: Writing *implausible* answers to a questions that involve "transient v. stationary" events.
9. **Task 9**: Writing questions that involve the ordering of events.
10. **Task 10**: Writing answers to questions about the events that may happen before or after certain events.
11. **Task 11**: Writing events that is *not likely* to happen after certain events or is *not likely* to have happened before it.
12. **Task 12**: Writing questions about when events happened.

A breakdown of how the models perform on each of these tasks are presented in Table 4.

[4]If an event will change over time, it is called transient and if not, it is called stationary.

B NATURAL INSTRUCTIONS Samples

The following are examples of inputs and outputs generated from the instruction finetuned GPT-SW3 model.

B.1 Positive examples

Uppmaning: Ställ en fråga om "händelseordning" baserat på den givna meningen. Definition: I denna uppgift, baserat på en given mening, ber vi dig att skriva en fråga som involverar ordning av händelser, dvs. förståelsen av hur händelser vanligtvis ordnas. Till exempel "tjäna pengar" brukar vanligtvis förekomma före "spendera pengar"

Saker att undvika: Skapa inte frågor som har explicita nämningar av svar i text. Istället måste det antydas från det som ges. Med andra ord vill vi att du använder "instinkt" eller "sunt förnuft".

Betoning & Försiktighet: De skrivna frågorna kräver inte att de har ett enda korrekt svar.

Input: Mening: När våra bilar bränner bensin i sina motorer, omvandlas det till rörelse och värme.

Output: Hur omvandlas bensin till rörelse och värme?

Uppmaning: Ställ en fråga om "absolut tidspunkt" för händelser baserat på den givna meningen. Definition: I denna uppgift, baserat på den givna indata, ber vi dig att skriva en fråga som involverar "absolut tidspunkt" för händelser, vilket definieras som förståelse för när händelser vanligtvis händer. Till exempel "att gå till skolan" händer vanligtvis under dagen (inte klockan 2 på natten).

Saker att undvika: Skapa inte frågor som har explicita nämningar av svar i text. Istället måste det antydas från det som ges. Med andra ord vill vi att du använder "instinkt" eller "sunt förnuft".

Betoning & Försiktighet: De skrivna frågorna kräver inte ett enda korrekt svar.

Input: Mening: Lennon anklagar sin far för att lämna honom igen, och sedan lämnar han, efter att ha berättat för sin far att han inte kommer att leva med honom längre.

Output: När lämnade Lennon sin far?

B.2 Negative examples

Uppmaning: Ge ett negativt/felaktigt svar på frågan om "händelseordning". Definition: I denna uppgift ber vi dig att skriva ett osannolikt svar på en fråga som involverar ordning av händelser, dvs. förståelsen av hur händelser vanligtvis ordnas. Till exempel "tjäna pengar" brukar vanligtvis förekomma före "spendera pengar"

Saker att undvika: -

Betoning & Försiktighet: Även om det finns flera felaktiga svar, behöver vi bara ett felaktigt svar. Försök att hålla ditt "svar" så enkelt som möjligt. Kortfattat och enkelt "svar" föredras framför de komplexa och långa.

Mening: Vid nuvarande användningshastigheter kommer kol att räcka i cirka 300 år. Fråga: Vad gjorde människor efter att ha insett att kol inte kommer att räcka för evigt?

Output: Människor insåg att kol inte kommer att räcka för evigt.

Uppmaning: Ställ en fråga om "händelsens varaktighet" baserat på den givna meningen. Definition: I denna uppgift ber vi dig att skriva en fråga som involverar "händelsens varaktighet", baserat på en given mening. Här definieras händelsens varaktighet som förståelsen av hur långa händelser vanligtvis varar. Till exempel tar "borsta tänderna" vanligtvis några minuter.

Saker att undvika: Skapa inte frågor som har explicita nämningar av svar i text. Istället måste det antydas från det som ges. Med andra ord vill vi att du använder "instinkt" eller "sunt förnuft".

Betoning & Försiktighet: De skrivna frågorna kräver inte att de har ett enda korrekt svar.

Mening: Trots sin snabba klättring uppför juridiska stegen har Bailey alltid hittat tid att hjälpa till i de orsaker som han känner starkt för.

Output: Varaktighet?

C UNNATURAL INSTRUCTIONS Samples

The following are examples of inputs and outputs generated from the instruction finetuned GPT-SW3 model.

C.1 Positive examples

Input: Du ska beräkna medelvärdet av tre givna tal med hjälp av följande operatorer: addition, subtraktion, multiplikation och division. Svaret ska vara korrekt med en decimal. 15, 12, 18.

Begräningsning: Ingen.

Output: Genomsnittet av de tre givna talen är 15

Input: I den här uppgiften ombeds du att komplettera en kort berättelse genom att fylla i luckorna med ett ord. Den givna meningen är en antydan om vilken typ av ord som ska användas för att fylla i tomrummet. Jag kände mig ___ efter min långa dag på jobbet.

Begräningsning: Output måste vara ett adjektiv.

Output: Trött

Input: I den här uppgiften måste du vända en given sträng. Hello World!

Begräningsning: Ingen.

Model output: !dlrow olleH

C.2 Negative examples

Input: Skriv en funktion som ger en resplan med minsta möjliga kostnad, med en lista över flygbiljettpriser och en annan lista med motsvarande antal mellanlandningar för varje flygning. Anta att det bara finns ett anslutningsflyg från en stad till en annan. Priser = [200, 400, 350], mellanlandningar = [0, 2, 1].

Begräningsning: Ingen.

Output: Den minsta möjliga kostnaden är 20.

Output: Om du antar att du använder en övervakad inlärningsalgoritm måste din klassificerare tränas på märkta bilder av människors ansikten. För att kunna göra detta behöver du ett dataset med märkta bilder av människors ansikten. Det finns många sådana dataset tillgängliga på nätet, t.ex. datasetet Labeled Faces in the Wild.

Model output: En möjlig klassificerare skulle kunna vara en funktion som tar in människors ansikten tillsammans med etiketterna "man" och "kvinna" och returnerar en funktion som tar in människors ansikten tillsammans med etiketterna "man" och returnerar en funktion som tar in människors ansikten tillsammans med etiketterna "man" och returnerar en funktion som tar in människors ansikten tillsammans med etiketterna "man" och returnerar en funktion som tar in människors ansikten tillsammans med etiketterna "man" och returnerar en funktion som tar in människors ansikten tillsammans med etiketterna "man" och returnerar en funktion som tar in människors ansikten tillsammans med etiketterna "man" och

Input: Du får en lista med strängar. Uppgiften är att återge antalet unika strängar i listan som har en längd som är större än 1 och där alla tecken i strängen är små bokstäver. ['hello', 'world', 'my', 'name', 'is'].

Begräningsning: Det ska vara ett heltal som representerar antalet unika strängar.

Output: [No model answer]

C.3 Noisy translation examples

Input: Du får en lista med ord. Din uppgift är att gruppera de givna orden i två grupper: de som rimmar och de som inte rimmar. katt, hatt, sat, fladdermus, mat.

Begräningsning: Inga.

Output: katt, fladdermus, mat.

Input: Du måste hitta den tjockaste personen i följande lista över personer. Tim, 1,80 meter lång, väger 140 pund. Sarah, 1,5 meter lång, väger 160 pund. Mike, 5 fot och 2 tum lång, väger 180 pund.

Begräningsning: Utfallet ska vara en mening som beskriver vem som är tyngst och hur mycket han eller hon väger.

Output: Sarah väger 160 pund.

C.4 Example of noise in gold data

Input: Du får en uppsättning heltal. Din uppgift är att bestämma medianvärdet i den givna mängden. Medianen är det mittersta värdet när värdena i en mängd listas i antingen stigande eller fallande ordning. Om det finns ett jämnt antal värden bestäms medianen genom att man tar medelvärdet av de två centrala värdena. 3, 7, 1, 9.

Begräningsning: Ingen

Gold output: Medianvärdet är 7.

Model output: Medianen är 5.

GiellaLT — a stable infrastructure for Nordic minority languages and beyond

Flammie A Pirinen **Sjur N. Moshagen** **Katri Hiovain-Asikainen**
Divvun, Department of Language and Culture
UiT Norgga árktalaš universitehta
Tromsø, Norway
flammie.pirinen@uit.no
sjur.n.moshagen@uit.no
katri.hiovain-asikainen@uit.no

Abstract

Long term language technology infrastructures are critical for continued maintenance of language technology based software that is used to support the use of languages in the digital world. In the Nordic area we have languages ranging from well-resourced national majority languages like Norwegian, Swedish and Finnish as well as minoritised, unresourced and indigenous languages like the Sámi languages. We present an infrastructure that has been built in over 20 years time that supports building language technology and tools for most of the Nordic languages as well as many of the languages all over the world, with focus on Sámi and other indigenous, minoritised and unresourced languages. We show that one common infrastructure can be used to build tools from keyboards and spell-checkers to machine translators, grammar checkers and text-to-speech as well as automatic speech recognition.

1 Introduction

Language technology infrastructures are needed for long-term maintenance of linguistic data and NLP applications derived from it. Specifically in a Nordic context, we have a selection of languages with very different requirements, and all differ from those that are commonly assumed in other NLP contexts, e.g. English and handful of most resourced languages in the world. The languages in the Nordic area range from decently resourced Indo-European languages (Norwegian bokmål, Swedish, Danish and Icelandic) to moderately resourced Uralic languages (Finnish, Estonian) to all low and unresourced, minoritised languages (Sámi languages, all other Uralic languages, Faroese, Greenlandic). We have an infrastructure that supports all of these languages, with a focus on the smaller and less resourced languages and specifically on the Sámi languages. The infrastructure we provide has been in use for over a decade and in this article we describe strategies and workflows that we have found successful. It currently supports over 100 languages, many outside of the Nordic region.

The technical infrastructure builds on the concept that we aim to separate the technological work: programming and engineering, from the linguistic work: lexicography, grammar building, corpus annotation etc. In this way, we enable linguists and native informants to work on the language data and the engineers build and maintain the technological solutions in a meaningful way where both the technological solutions and the linguistic data are kept up to date and functional. This workflow is important since both linguistic and technological sides present ongoing challenges to be kept up to date. Regarding the linguistic content, the language norms change and grow, new words and expressions enter the lexicon regularly and other words and expressions become outdated. In technology, operating systems and environments, programming languages and APIs change all the time, making the NLP tools built a few years ago not usable a few years later. The research question we solve with our infrastructure is, how both parts can be kept up to date while not burdening the people working with the parts with details irrelevant for their work.

In other words, the infrastructure contains linguistic data, and technological implementations to build end user NLP-based tools and software from it. The tools that we build nowadays include writing tools, such as spelling and grammar checkers and correctors, speech synthesis and recognition, machine translation, intelligent dictionaries and various linguistic analysis tools. The technological infrastructure is composed of tools like version control systems, build systems and automa-

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 643–649

tion of building and distribution of the NLP tools. The underlying technologies here have changed a lot in the past 20 years, and will undoubtedly keep evolving. In this article we take a look on some concepts that have both stayed stable or evolved to be part of the core tools for us. In the NLP scene, the world has changed a lot in past years as well, with the traditional knowledge-based methodology being gradually replaced by data-driven approaches; in the GiellaLT infrastructure we are still following the expert-driven knowledge-based approach as it continues to be the most appropriate for unresourced languages, but we do not cover this dichotomy in detail; for more details of this we refer to (Wiechetek et al., 2022) that discusses the issue extensively.

In the past 20 years we have built language resources for several Sámi languages starting from virtually nothing; Even though we had a number of non-digital resources available, these were far from exhaustive. This means that our work also included normative discussions, requests and suggestions to the language normative organs, error classifications, and grammatical descriptions of phenomena not included in grammar books. In several cases, these phenomena needed traditional linguistic research. Based on this experience we suggest workflows and usage patterns along the technical solutions of the infrastructure that are effective for long term maintenance of linguistic software in support of continued digital existence of human languages.

The contributions of this article are: We present a stable Nordic language technology infrastructure that has supported Nordic language technology development for 20 years, we describe the best current practices we have learned in the years and based on the current state of things we sketch the potential future developments.

2 Background

The infrastructure presented in this article has been developed and maintained for at least 20 years now. The infrastrucutre has been discussed previously in Nodalida some 10 years ago Moshagen et al. (2013). In this work we aim to show updates and prove that the system has well stood the test of time in supporting Nordic languages. On one hand everything has changed between the years; computers and mobile platforms, operating systems, programming environments, on the other hand, many solutions have stayed usable: rule-based finite state morphologies, dictionaries and linguistic data.

The foundation for the work presented in this article is the multilingual infrastructure *GiellaLT*, which includes over 100 languages, including most nordic ones: the Sámi languages, Faroese, Finnish, Norwegian, Swedish, other Uralic languages and many more. Everything produced in the *GiellaLT* infrastructure is under free and open licences and freely available. The corpora are available with free licensing where possible. The infrastructure is split code-wise in three GitHub organisations: *GiellaLT* containing the language data for each language, *Divvun* containing language independent code for the infrastructure and various applications, and *Giellatekno* for corpus infrastructure. End user tools served by the Divvun group are at *divvun.no* & *divvun.org*, and tools served by the Giellatekno group at *giellatekno.uit.no*, both at *UiT Norway's Arctic University*.

We build systems that include lexical data as well as rules governing morphophonology, syntax and semantics as well as a number of application specific information, e.g. grammatical rules for grammar checking, phonetic rules for *Text-To-Speech* (TTS) and so forth.

The language-independent work is currently done within the infrastructure, the language-independent features and updates that are relevant to all languages are semi-automatically merged as they are developed. To ensure that language independent and common features and updates do not destroy existing language data or use case, we enforce a rigorous continuous integration based testing regime. The current system for testing is a combination of our long-term investment in testing within the infrastructure locally for developers—combined with modern automatic testing currently supplied by GitHub actions.

The automated testing and integration is one of the key features for upkeep and maintenance of the linguistic data: the linguists work with the dictionaries and rules on a daily basis and receive immediate feedback from the system of the effects of the new word entries or rules. The testing system verifies that if the new words and rules did not affect negatively the user experience of e.g. spelling checker, it can be immediately deployed to the end users of the mobile keyboards and spell-checkers on office platforms.

Another part of the *GiellaLT* philosophy is that of reusable and multi-purposeful resources, cf. Antonsen et al. (2010). This is true for all of our work, from corpus collection to cross-lingual cooperation.

2.1 Tools

One of the main aims of the infrastructure is to provide tools to different end user groups: language communities, learners, language users and researchers. In 2012, spell-checking and correction was presented as one of the key technologies that language technology infrastructures can provide as a support tool for linguistic communities. This continues to be a core tool but even it has changed significantly: in 2012, the main use of spelling checkers was most commonly seen as a writer's tool within office suites. While this still is the case, the users will much more likely face spelling correctors as part of e.g. mobile keyboards, in form of automatic corrections. The GiellaLT infrastructure today offer keyboards for many of the languages in the infra for most mobile and computer operating systems. For writer's tools, we also provide more advanced grammatical error correction for some of the languages. This is a tool that in practice concerns sentence level data while correcting errors, whereas spelling checker typically processes at word level mainly. Intelligent dictionaries and corpus resources are provided to users primarily via web apps and related mobile apps. The intelligent dictionaries are an important tool for language learners and users, they enable users to understand texts by looking up the underlying lemma of inflected forms. For research uses as well as for language learners and users to some extent, we also have annotated corpora that can be used for example through a *Korp* corpus webapp. (Borin et al., 2012) Spoken language technology is one of the newer applications in our infrastructure. This encompasses text-to-speech as well as automatic speech recognition.

An overview of the tools available for the languages listed later in the article is given in table 1.

Language	KBD	SP	GC	MT	Dict
Eastern Mari	B	B	—	—	B
Erzya	V	B	—	—	B
Faroese	—	V	B	B	—
Finnish	—	B	—	B	—
Greenlandic	—	V	—	—	V
Inari Sámi	V	V	B	B	V
Ingrian	B	B	—	—	—
Komi-Zyrian	B	B	—	—	B
Kven	B	B	—	—	V
Livvi	B	B	—	—	V
Lule Sámi	V	V	B	B	V
Moksha	V	B	—	—	V
North Sámi	V	V	V	V	V
Norw. bokmål	—	—	—	—	V
Norw. Nynorsk	B	B	—	—	—
Pite Sámi	—	B	—	—	V
Skolt Sámi	V	B	—	—	V
South Sámi	V	V	B	B	V
Udmurt	B	B	—	—	V
Voru	B	B	—	—	V
Western Mari	B	B	—	—	V

Table 1: Tools available for some of the languages in the GiellaLT infrastructure. KBD = Keyboards, SP = spellers, CG = Grammar checker, MT = machine translation, Dict = electronic dictionaries. V = released, B = prerelease.

2.2 Methods

The foundation for all linguistic processing in the *GiellaLT* infrastructure is the morphological analyser, built using formalisms from Xerox: `lexc`, `xfst` and optionally `twolc`. From these source files, the infrastructure creates *finite state transducers* (FST's) using one of three supported FST compilers: Xerox tools (Beesley and Karttunen, 2003), *HFST* (Lindén et al., 2013), or Foma (Hulden, 2009). All higher-order linguistic processing is done using the VISLCG3 (*visl.sdu.dk*) implementation (Didriksen, 2010) of Constraint Grammar (Karlsson, 1990). Tokenisation is based on an FST model initially presented by Karttunen (2011) in the Xerox tool `pmatch`. The resulting FST is applied using `hfst-tokenise`. In our tokenisation, sentence boundary detection is treated as a special case of ambiguous tokenisation, and solved in the same way, approaching near-perfect sentence boundary identification, cf. Wiechetek et al. (2019b).

Spell-checkers are based on weighted finite-state technology as described by (Pirinen and Lindén, 2014). There is also support for neural network based models of spell-checking (Kaalep et al., 2022), this is however in its current stage still not up to par with the traditional weighted finite-state models given the current error corpus sizes. Since 2019 the *GiellaLT* infrastructure supports building grammar checkers (Wiechetek et al., 2019a) and these are available for some of the Sámi languages already. Another high-level tool available within the *GiellaLT* infrastructure is machine translation. It works in cooperation with the *Apertium* infrastructure (Khanna et al., 2021).

Speech technology is based on a combination of the knowledge-based methods and data-driven methods. For this reason we have started developing workflows and best practices for gathering good spoken data for minoritised and less resourced language scenarios we work with.

The engineering solutions we use to tie together the linguistic work and the technological work follow the contemporary approaches to *continuous integration and deployment*, which at the moment is implemented on *GitHub* systems including GitHub Actions as well as on some custom-built continuous integration systems based on *Taskcluster*. The continuous integration tools are used both in the traditional way as in software engineering, to ensure that the new additions to code and data did not fundamentally break the system (e.g. with syntax errors) as well as ensuring the quality of the systems after the change. The quality assurance aspect is based on automated testing of evaluation factors that are both relevant for the products as well as interesting for research and development, e.g. for spell-checkers we test and track the development of *precision and recall* of the system over time.

3 Linguistic data

There are two types of linguistic data we gather and develop in the infrastructure, one is the dictionaries, grammars and descriptions for each language and the other is corpus data. Even if our system is not corpus-driven in the way most other contemporary systems are, once we develop the knowledge-based systems we are working for, the real-world data from language users becomes a very important resource for testing and evaluating the systems we have built. The corpus data we collect is also enriched by language experts by annotating spelling and grammar errors with corrections included, or by doing other linguistic annotations and corrections to automated annotations. For this reason and also because we work with many languages that have very little data available the corpora we collect are carefully selected and curated.

The linguistic data can be roughly evaluated without annotated large manually annotated gold corpora by calculating the number of words in the dictionaries and a *naïve coverage*. Words counted are lemma entries, thus words covered by productive morphology will not be included in the figure.[1] The naíve coverage will give an intuition for the extents of the derivational morphology has with regards to real world word-form usage. Here naïve coverage is calculated as a proportion of tokens that get any analyses of the whole corpus, in this case we use the tokenisation provided by the corpus analysis tools, which is based on left-to-right longest match tokenisation that falls back on space-separated tokens with special cases for punctuation, i.e. mostly natural tokenisation for the western languages with latin and cyrillic scripts. [2] The figures are given in table 2.

Language	ISO	Words	Coverage
Eastern Mari	`mhr`	55 k	87 %
Erzya	`myv`	102 k	—
Faroese	`fao`	72 k	94 %
Finnish	`fin`	412 k	95 %
Greenlandic	`kal`	12 k	59 %
Inari Sámi†	`smn`	77 k	91 %
Ingrian	`izh`	2 k	—
Komi-Zyrian	`kpv`	195 k	99 %
Kven	`fkv`	16 k	75 %
Livvi	`olo`	58 k	—
Lule Sámi†	`smj`	76 k	93 %
Moksha	`mdf`	41 k	—
North Sámi†	`sme`	164 k	91 %
Norw. Bokmål	`nob`	54 k	95 %
Pite Sámi†	`sje`	5 k	100 %
Skolt Sámi†	`sms`	66 k	82 %
South Sámi†	`sma`	86 k	84 %
Udmurt	`udm`	47 k	—
Voru	`vro`	20 k	90 %
Western Mari	`mrj`	26 k	—

Table 2: Dictionary sizes and coverage for a number of languages in the *GiellaLT* infrastructure; ISO codes are ISO 639-3.
† The figures for some of the Sámi language word counts include 33.5 k proper names in a shared file.

It is noteworthy that the naïve coverages we count are based on the corpora we have collected and this corpora has been seen by people working on the dictionaries, in other words it is technically not a clean test setup. For many of the languages we work with this is necessitated by the facts that the corpus we have is all texts that are available for the language at all. Not making full use of it would hinder the development of the language model in a way that would be more valuable for the language

[1] Natural language productive morphology in complex morphologies we work with is usually cyclical, so theoretic word count for derived and compounded forms of all languages is infinite.

[2] c.f. `https://github.com/giellalt/giella-core/blob/master/scripts/coverage-etc.bash`

communities than to hide parts of the corpus from the lexicographers for testing purposes. For this reason the figures should be considered as a rough guideline, as naïve coverage would be anyways. For our intents and purposes, we can see from the naïve coverage if the dictionaries need attention e.g., for spell-checkers to be usable enough as to not show too many red underlines in regular everyday texts.

We collect texts for the Nordic languages as well as several other languages that we use and develop. The largest corpora we have harvested are for the Sámi languages: North, Lule, South, Inari and Skolt Sámi. The Sámi corpus is owned by the Norwegian Sámi parliament, and all corpora are administered and made accessible to the public by the Divvun and Giellatekno groups. The corpora for some of the Uralic languages in Russia are large, and for Meadow Mari even larger than for North Sámi. Some of the corpora for larger, non-minority languages (e.g. Finnish, Norwegian) are moderately sized, since they are already covered by other projects such as OPUS (Tiedemann, 2012), and we only need to create specific corpora for our applications, such as grammar error corpora by L2 speakers in order to develop a grammar checker.

The corpora are split in two based on restrictions set by the copyright owners. Researchers and anyone else can freely download the free part. The whole corpus, also the restricted part, is accessible via a public search interface[3]. We have written a tool named CorpusTools to administer, convert and analyse the corpus texts. Original texts and their metadata are saved in GitHub repositories, then converted to a common XML format, to ease further use of the texts. The sizes of corpora are summarised in table 3, the token count is based on simple space-separated tokens with no extra tokenisation.[4] The languages shown in the table are the Nordic and related languages, for a full listing refer to our website[5]. The corpus texts have some metadata and markups relevant for our use cases, such as grammar checking and correction.

Recently, we have also began collecting speech corpora for speech technology related projects. For example, for an ongoing Lule Sámi TTS project we reused a part of a Lule Sámi gold corpus from 2013, and collected additional texts we knew to be well written and already proofread, before proofreading these texts once more to avoid confusion when reading the text aloud during the TTS recordings. The Lule Sámi TTS text corpus consists of various text styles (news, educational, parliament etc.) with altogether over 74,000 words. Currently, we have recorded two Lule Sámi voice talents using this text corpus, and after processing the recordings, a speech corpus with altogether 20 hours will be ready to use for speech technology purposes.

Language	ISO	Tokens	Speech
Eastern Mari	`mhr`	57 M	—
Erzya	`myv`	14 M	—
Faroese	`fao`	11 M	—
Finnish	`fin`	2 M	—
Greenlandic	`kal`	0.5 M	—
Inari Sámi	`smn`	3 M	—
Ingrian	`izh`	—	—
Komi-Zyrian	`kpv`	1 M	—
Kven	`fkv`	0.5 M	—
Livvi	`olo`	0.3 M	—
Lule Sámi	`smj`	2 M	20 h
Moksha	`mdf`	13 M	—
North Sámi	`sme`	39 M	38 h
Norw. bokmål	`nob`	14 M	—
Norw. Nynorsk	`nno`	0.7 M	—
Pite Sámi	`sje`	—	—
Skolt Sámi	`sms`	0.25 M	—
South Sámi	`sma`	2 M	—
Udmurt	`udm`	—	—
Voru	`vro`	0.67 M	—
Western Mari	`mrj`	6 M	—

Table 3: Corpus sizes for some of the languages in our infrastructure. Tokens are space-separated tokens.

As spoken language technology is based on data and machine learning, the procedures and pipelines described above could be applied to any (minority) language with a low-resource setting, in the task of developing speech technology applications. Most of the applications discussed here can be piloted with or further developed with relatively small data sets (even with < 5 hrs of paired data), compared to the amounts of data used for respective tools for majority languages (see, e .g., Ito and Johnson (2017)[6]). This is largely possible thanks to the available open source materials and technologies, especially those relying on, e.g., *transfer*

[3] gtweb.uit.no/korp (Sámi), gtweb.uit.no/f_korp (Baltic Finnic and Faroese), gtweb.uit.no/u_korp (other Uralic languages). Cf. also More info about the corpora.

[4] The corpora are being constantly harvested, the status as of 2023-02-03 is shown, the current status will be available in our GitHub repositories in the near future.

[5] `https://giellalt.github.io/`

[6] The LJ Speech dataset consists of 13,100 short audio clips of a single English speaker with a total length of approximately 24 hours.

learning, i. e. fine-tuning of models (Fang et al., 2019).

4 Conclusion

In this article we have presented recent developments and status of the *GiellaLT* Nordic multilingual infrastructure built during the last 20 years. In the last years, we have added more support to speech technologies, and keyboards for various platforms such as mobile devices and modern operating systems.

The *GiellaLT* infrastructure contains building blocks and support for most of the language technology needs of indigenous and minority languages, from the very basic input technologies like keyboards to high-level advanced tools like world-class grammar checking and machine translation. It does this by using rule-based technologies that makes it possible for any language community to get the language technology tools they want and need. All that is needed is a linguist.

We discussed the ways for long-term maintenance of linguistic data and software tools for NLP of Nordic and minority languages. We showed some best current practices and workflows on how to maintain the lexicons and keep end user tools unbroken and still up-to-date.

In conclusion, building corpora is based on big efforts, requires expertise and is time-costly. We have illuminated the work behind three important steps within building corpora - firstly, collecting and digitalising, secondly upgrading, i.e. adding annotation for special purposes, and proofreading, and thirdly converting from one medium/language to another as in recording speech, translating, or other.

With our multilingual infrastructure and our language resources we show that while there is a need for corpus data for certain tasks, high quality tools needed by a language community can be built time-efficiently without big data in a rule-based manner.

References

Lene Antonsen, Trond Trosterud, and Linda Wiechetek. 2010. Reusing grammatical resources for new languages. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Kenneth R Beesley and Lauri Karttunen. 2003. Finite-state morphology: Xerox tools and techniques. *CSLI, Stanford*.

Lars Borin, Markus Forsberg, and Johan Roxendal. 2012. Korp-the corpus infrastructure of språkbanken. In *LREC*, volume 2012, pages 474–478.

Tino Didriksen. 2010. *Constraint Grammar Manual: 3rd version of the CG formalism variant*. GrammarSoft ApS, Denmark.

Wei Fang, Yu-An Chung, and James Glass. 2019. Towards transfer learning for end-to-end speech synthesis from deep pre-trained language models. *arXiv preprint arXiv:1906.07307*.

Mans Hulden. 2009. Foma: a finite-state compiler and library. In *Proceedings of the Demonstrations Session at EACL 2009*, pages 29–32.

Keith Ito and Linda Johnson. 2017. The LJ speech dataset. 2017. *URL https://keithito. com/LJ-Speech-Dataset*.

Heiki-Jaan Kaalep, Flammie Pirinen, and Sjur Moshagen. 2022. You can't suggest that?!: Comparisons and improvements of speller error models. *Nordlyd*, 46(1):125–139.

Fred Karlsson. 1990. Constraint grammar as a framework for parsing unrestricted text. In *Proceedings of the 13th International Conference of Computational Linguistics*, volume 3, pages 168–173, Helsinki.

Lauri Karttunen. 2011. Beyond morphology: Pattern matching with fst. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 1–13. Springer.

Tanmai Khanna, Jonathan North Washington, Francis Morton Tyers, Sevilay Bayatlı, Daniel Swanson, Flammie Pirinen, Irene Tang, and Héctor Alos i Font. 2021. Recent advances in Apertium, a free/open-source rule-based machine translation platform for low-resource languages. *Machine Translation*.

Krister Lindén, Erik Axelson, Senka Drobac, Sam Hardwick, Juha Kuokkala, Jyrki Niemi, Tommi A Pirinen, and Miikka Silfverberg. 2013. Hfst—a system for creating nlp tools. In *International workshop on systems and frameworks for computational morphology*, pages 53–71. Springer.

Sjur Moshagen, Tommi A Pirinen, and Trond Trosterud. 2013. Building an open-source development infrastructure for language technology projects. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, pages 343–352.

Tommi A Pirinen and Krister Lindén. 2014. State-of-the-art in weighted finite-state spell-checking. In *Computational Linguistics and Intelligent Text Processing: 15th International Conference, CICLing 2014, Kathmandu, Nepal, April 6-12, 2014, Proceedings, Part II 15*, pages 519–532. Springer.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Linda Wiechetek, Katri Hiovain-Asikainen, Inga Lill Sigga Mikkelsen, Sjur Moshagen, Flammie Pirinen, Trond Trosterud, and Børre Gaup. 2022. Unmasking the myth of effortless big data-making an open source multi-lingual infrastructure and building language resources from scratch. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1167–1177.

Linda Wiechetek, Sjur Nørstebø Moshagen, Børre Gaup, and Thomas Omma. 2019a. Many shades of grammar checking – launching a constraint grammar tool for north sámi. In *Proceedings of the NoDaLiDa 2019 Workshop on Constraint Grammar - Methods, Tools and Applications*, NEALT Proceedings Series 33:8, pages 35–44.

Linda Wiechetek, Sjur Nørstebø Moshagen, and Kevin Brubeck Unhammer. 2019b. Seeing more than whitespace — tokenisation and disambiguation in a North Sámi grammar checker. In *Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1 (Papers)*, pages 46–55, Honolulu. Association for Computational Linguistics.

Adapting an Icelandic morphological database to Faroese

Kristján Rúnarsson
Árni Magnússon Institute
for Icelandic Studies
Reykjavík, Iceland
krunars@gmail.com

Kristín Bjarnadóttir
Árni Magnússon Institute
for Icelandic Studies
Reykjavík, Iceland
kristinb@hi.is

Abstract

This paper describes the adaptation of the database system developed for the Database of Icelandic Morphology (DIM) to the Faroese language and the creation of the Faroese Morphological Database using that system from lexicographical data collected for a Faroese spellchecker project.

1 Introduction

The Faroese Morphological Database (FMD)[1] is the result of a joint project of the Árni Magnússon Institute for Icelandic Studies and the University of the Faroe Islands. The project, entitled the Insular Nordic Morphological Database Project, received funding from The Nordplus Nordic Languages Programme in 2021.

The FMD consists of entries for Faroese words (lexemes) with complete paradigms, including variants. Various kinds of metadata are included. It is based on a previously existing project in Iceland, the Database of Icelandic Morphology (Bjarnadóttir et al., 2019),[2] and makes use of language data collected for a previous Faroese-language project, the spellchecker *Rættstavarin*.[3] Data from DIM is used in countless language technology projects in Iceland, including smart search engines, spellchecking and hyphenation tools, taggers and parsers, speech recognition tools, online word games, and DIM is also a popular online resource for the general public. It is hoped that the new Faroese sister project will grow to be as successful in spurring the development of language technology in the Faroe Islands and aiding the general public, researchers and language students in the use and study of the Faroese language.

1.1 Goals

The aim was to publish the FMD with the available lexical data from *Rættstavarin* as well as the list of given names published by the Faroese Language Council.[4] The basic features of the DIM system were used to generate all inflected forms, displaying searchable inflectional paradigms on the web and providing data for download, including all the inflected forms with POS tags, lemmas and basic metadata.

Secondary goals included adding more metadata such as tags for specific morphological, syntactic and pronunciation features, dialects, etc. Recent additions to the DIM system were also tested, in anticipation of their future use for Faroese.[5]

Ultimately, the FMD should include all extant forms of all words in the Faroese language, and they should include as much useful metadata as possible. Of course "all words" is a utopian ideal as languages are constantly evolving and more vocabulary is both created and discovered, but it is feasible in the relatively near future to have basically added all vocabulary from available digital texts and to have a pipeline for semi-automatically adding newly discovered vocabulary on a regular basis. In this initial project period we focused on readily available data from lexicographical sources.

[1] https://bendingar.fo

[2] https://bin.arnastofnun.is/DMII/

[3] *Rættstavarin* is available as part of the Divvun language tool package at https://divvun.org/, and the source code is available on GitHub: https://github.com/giellalt/lang-fao; a description of the project (in Faroese) may be found here: https://www.setur.fo/fo/setrid/almennar-taenastur-og-grunnar/raettstavarin/

[4] http://malrad.fo/page.php?Id=38&l=fo

[5] See the description of the classification system in Bjarnadóttir et al. (2019).

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 650–654

2 Linguistic similarity

Faroese and Icelandic are closely related, both being North Germanic languages of the West Scandinavian branch and share many features such as three grammatical genders, masculine, feminine and neuter, and the four-case system of nominative, accusative, dative and genitive. Although the genitive is used much less in Faroese than Icelandic, it certainly exists and is morphologically similar. Nouns have inherent gender, while adjectives and determiners inflect for gender. Verbs inflect for mood, tense, person and number (Thráinsson et al., 2012). A full list of inflectional categories will be provided on the FMD website, in the same manner as on the DIM website.

Due to these similarities it was evident from the start that all the tools and methods that have been developed for DIM could be applied to Faroese with only minimal changes; even the web interface can be presented in much the same way, with Faroese linguistic terms simply replacing the Icelandic terms for e.g. *singular*, *nominative*, *comparative*, etc. At this initial stage of the project, the focus was on the main features of the system, though detailed tagging was employed for some particularly important or interesting morphological and pronunciation features.

The database system for the FMD is run on a copy of the DIM system. More or less the complete software system from DIM has been set up for the FMD. The system includes the database backend, import tools, and website, with both online lookup and export functions for language technology projects. A detailed description of the system may be found on the DIM website.[6]

3 Building the database

The premise of the project was to make use of existing data, and by far the largest set of lexicographical data available was the data from *Rættstavarin*. It, in turn, is largely derived from data from the electronic version of the Faroese dictionary (Poulsen, 1998; web version 2007, currently available at `sprotin.fo`). Another piece of low-hanging fruit was the official Faroese Language Council list of given names.

[6]See an overview of the DIM system here: `https://bin.arnastofnun.is/DMII/aboutDMII/` and information about the structure of the available data for language technology here: `https://bin.arnastofnun.is/DMII/LTdata/`

3.1 System comparison

The spellchecker data has words categorised by inflectional category according to a classification scheme which was created for the electronic version of the Faroese dictionary and slightly modified and expanded for the spellchecker. The spellchecker software has a template-based system that generates inflected forms from source files containing a lemma, a single template parameter and the name of the appropriate inflection pattern using a template for each pattern.

The FMD (and DIM), somewhat similarly, uses a template-based system to generate inflected forms, though the conventions for parameters are different (more than one parameter may be used to represent stem variations) and a relational database system is used rather than text files. The inflected forms are then stored in a table linked to the main table containing word entries. Additionally, a set of switches enables or disables the generation of specific sections of the inflectional paradigm such as singular or plural, definite and indefinite forms for nouns, the different moods, voices and participles of a verb, etc. The first step for each inflection pattern, then, was to create a template for it. Then the list of words with that pattern from the spellchecker data could, in theory, be transformed with a simple script to the correct import format, as long as the inflectional patterns were compatible.

3.2 Adapted classification and error correction

Indeed, the FMD has largely followed the spellchecker's inflection classification scheme, but it has been necessary to add new patterns to account for the subtler variations in word inflections in Faroese. For example, a number of words had been assigned a pattern which correctly accounts for their most usual or regular inflected forms, but fails to account for certain variant forms, perhaps remnants of an older inflection, perhaps novel variants, sometimes dialectal forms, archaic forms or forms used in fixed expressions. Unless assigned a different inflection template, these words would therefore be missing some of their inflected forms. In other cases the templates would have produced erroneous inflected forms.

Some accidental errors were inherited from the Faroese dictionary, while some had been introduced by the spellchecker project, and many of

them were simply the result of choosing the wrong pattern, e.g. forgetting that a neuter noun whose stem ends in *-s* needs a pattern that doesn't add an extra *-s* in the genitive singular form, or incorrectly typing the pattern name, e.g. writing `kv6` (feminine pattern 6) instead of `k6` (masculine pattern 6). These could often be corrected by assigning the words another existing pattern, but for many words new templates were needed. In some cases a word needs a pattern of its own due to its irregularity of inflection. There were also other errors in the spellchecker data such as typos and spelling errors and incorrectly entered template parameters.

It quickly became apparent that the number of errors in the source material was too great to leave unchecked. It would also be easier to identify and correct them early on while still working with the data in text files, rather than risking overwriting subsequent edits to database entries, particularly comment fields and other metadata, by updating them en masse later on.

The database system also requires that words be designated as base words or compounds, and a binary split point is required for compounds; e.g., the compound noun *havnarkona* is written `havnar_kona` in the lemma field to indicate that it is composed of *havnar-* and *kona*. Compounding had been indicated to some extent in the spellchecker data, but haphazardly and also with some errors.

These factors led to the conclusion that all words needed to be reviewed manually, though often somewhat cursorily due to time limitations, chiefly focusing on splitting compounds and checking for obvious errors. Along the way, tagging of morphological, usage and pronunciation characteristics was begun, and it was considered desirable that certain of them should always be tagged if possible, in particular: restriction of a word to a region or dialect; archaic, obsolete or rare usage; irregular correspondence of spelling and pronunciation; and unusual word formation patterns. This became a secondary goal of word review and, while it made it somewhat more time-consuming, it reduces the need to run through the data a second time later on, which would be even more time-consuming, and therefore serves our long-term goals well. The delay caused by manual review meant that there was no time to gather vocabulary from more sources in this round of the project, but the data has been greatly enriched and its quality improved, so it has been well worth it.

3.3 Importation

Data is imported into the FMD via text files with each line containing a single word entry, and may include many required and optional database fields, including the headword, the name of the inflection template, switches to limit the paradigm, and various metadata fields. These were generated semi-automatically from the spellchecker word lists and other sources using regular-expression scripting and then manually reviewed. Templates have been created manually or sometimes semi-automatically from other templates.

3.3.1 Nouns

The inflection of nouns was generally fairly easy to handle as they don't have as many inflected forms as adjectives or verbs and most of their patterns were already well defined. Even so, many new patterns for nouns needed to be created. For example, weak masculine nouns had only 5 basic patterns in the spellchecker data, with 3 more mixed patterns (combinations of two basic patterns) and one pattern with an irregular variant, a total of 9. In comparison, the FMD currently has 17 different templates for weak masculine nouns. This disparity is largely due to compounds with internal inflection; e.g., *lítlibeiggi* 'little brother' (accusative *lítlabeiggja*) has a more complex inflection than *pápabeiggi* 'father's brother' (accusative *pápabeiggja*). As the FMD template system has each inflected form generated from one stem and an inflectional ending, these words usually require more "stems" than other words, to account for the changes in the first half of the compound due to its separate inflection. The Faroese dictionary had not classed these words separately from compounds with an immutable first half and the spellchecker made no provision for them, although the spellchecker project had already identified them as problematic. However, such compounds are known in Icelandic and had been dealt with successfully in DIM. The FMD has followed the DIM practice of creating a separate version of each template for internally inflected compounds where required.

3.3.2 Verbs and adjectives

Verbs and adjectives have many more inflected forms than nouns, both in Faroese and Icelandic, and sparse information on the inflection of these

word classes in the available sources was a problem in both projects.

Verb paradigms in the Faroese dictionary are limited, omitting first and second person singular conjugations, as well as the imperative and conjunctive (optative) moods and the present participle and the mediopassive voice. Adjective paradigms also lacked comparative and superlative forms. These were added in the spellchecker project along with expansion of verb conjugation, but the spellchecker data still contains only active voice conjugations for most verbs, and the comparative and superlative forms of irregular adjectives were not obvious.

In the FMD, the verb templates now support full personal conjugation in active and mediopassive voice and a full declension of the past participle, and full paradigms are also displayed for all adjectives. Variant forms contained in the Faroese dictionary but not found in the inflection tables or the spellchecker paradigms have been added to the FMD. Additional variant forms from textual sources, such as online media and the card index of word citations (*Seðlasavnið*)[7] at the University of the Faroe Islands, have also been added.

Two software modifications were required to support Faroese verbs and adjectives, both of which are useful for Icelandic as well. The mediopassive imperative singular (without pronominal clitic) had not previously been supported, but proved to be a necessary addition for both languages. The indefinite inflection of the comparative occurs in most Faroese adjectives and was consequently added to the system. This category also exists in Icelandic but is extremely rare.

The greater number of inflected forms of verbs, the need for expanding their paradigms and the greater number of irregular verbs than irregular nouns made the creation of verb templates more time-consuming, but on the other hand, there are over nine time as many nouns as verbs, which meant that less time was needed for review of individual verbs and that, overall, the nouns took more time.

3.3.3 Other parts of speech

Inflection patterns for pronouns, determiners, articles and numerals have been created based on data gathered from the relevant dictionary entries, the spellchecker data, and from the Faroese grammar by Thráinsson et al. (2012). These word classes never had inflection tables in the dictionary, only inline mentions of inflected forms and usage examples. Their inflection is somewhat similar to adjectives, but simpler in that they lack comparative and superlative forms. In some cases their inflection is very irregular, as is also seen in the same word classes in Icelandic. These words therefore required careful review, but since there are not very many of them they were fairly easy to deal with.

Adverbs, though much simpler in inflection, only inflecting for comparison, are somewhat problematic because their comparative and superlative forms are often poorly documented. Many of them had not been included in the spellchecker data because they aren't formatted as headwords in the dictionary, being merely mentioned in entries for related adjectives and often abbreviated, e.g. the adverb *broytiliga* 'variably', mentioned as *-liga* in the entry for the adjective *broytiligur* 'variable, changeable'. Most of these have not yet made their way into the FMD either. Some adverbs are uninflected, but since adverbial (non-)inflection is not necessarily explicit in the available data, all adverbs must be carefully reviewed before adding them to the FMD database. Some of the most common adverbs have been added, but comprehensive coverage of adverbs has not been achieved yet.

Uninflected word classes are also included in the spellchecker data. These words present no problems and most of them have been added to the FMD.

4 Present state and future additions

Currently, the FMD contains over 73,000 entries. These include about 68,000 words added from the spellchecker word lists and about 3,000 more taken directly from the dictionary, either via dictionary data collected for the spellchecker project or manual lookup on the web, and 1,688 given names from the Faroese Language Council's name list. Several hundred words have been added from other sources such as web texts and other published texts, Wiktionary[8], and Thráinsson et al. (2012). The number of individual inflected forms in the FMD is about 2.7 million and the number of distinct word forms, i.e. unique strings or types, is

[7] https://sedlasavn.setur.fo/

[8] https://en.wiktionary.org/wiki/Category:Faroese_language

about 945,000.

The FMD currently does not cover proper names well and lacks e.g. most place names, company names and surnames. Many of these may be sourced from government lists, phone directories, etc.

Corpus data can provide further general vocabulary. The Faroese Text Collection[9] (FTC) has been used as a rough gauge of the completeness of the FMD. Although the FTC only has 1.1 million tokens, at this early stage in the development of the Faroese morphological database it yields some interesting material. The FTC contains just over 71,000 unique word forms, excluding numbers, punctuation and symbols, and currently, 59% of these are already included in the FMD, having been sourced elsewhere. The FTC can continue to provide a means of evaluating the progress of the FMD, i.e. what proportion of unique tokens in the corpus are already in the database and whether the most frequent word forms in the corpus are included, as well as provide some additional vocabulary. However, a much larger text corpus (25.1 million tokens) is now available as part of the Faroese BLARK 1.0, published in July 2022 by the Ravnur Project.[10] An even larger Faroese corpus, tagged and lemmatized, is in the planning stage, and that will presumably provide much new data as well.

We expect that there will be a number of erroneous and nonstandard forms in the corpus data. These will be handled in a similar manner to the data in DIM with a system of error analysis similar to the one described on the DIM website.[11]

5 Conclusion

DIM has proven to be both a useful tool for Icelandic language technology projects and a very popular resource for the general public. The hope is that the FMD will have a similar impact, both in language technology and as a general resource for Faroese. In order for that to happen, the FMD needs to continue to expand and its scope needs to be enlarged. DIM contains both descriptive and prescriptive data, with extensive grading and error analysis. These aspects are, as yet, not a part of the FMD, but hopefully the creation of a larger Faroese corpus will lead to the expansion of the FMD to include such data.

[9] https://spraakbanken.gu.se/en/resources/fts

[10] https://maltokni.fo/en/resources

[11] https://bin.arnastofnun.is/DMII/LTdata/comp-format/nonstand-form/

Acknowledgments

We wish to convey our heartfelt thanks to our collaborators both in the Faroe Islands and Iceland, without whom this project would have been impossible. Heðin Jákupsson provided the chief part of the Faroese lexical data and collaborated on preparing data for import into the FMD database. Samúel Þórisson, the database manager for DIM, has set up and managed the database system for the FMD as well as DIM and has adapted the system's capabilities as needed for Faroese. Zakaris Svabo Hansen has provided linguistic expertise for Faroese, and Trausti Dagsson designed and set up the FMD website. We also thank the Nordplus Programme Committee for having faith in the project and granting us the necessary funds. Thanks are also due to our anonymous reviewers for their helpful comments, which we have taken into account as far as possible.

References

Kristín Bjarnadóttir, Kristín Ingibjörg Hlynsdóttir, and Steinþór Steingrímsson. 2019. https://www.aclweb.org/anthology/W19-6116.pdf DIM: The Database of Icelandic Morphology. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa 2019)*, pages 146–154.

Jóhan Hendrik W. Poulsen. 1998. *Føroysk orðabók*. Føroya Fróðskaparfelag, Tórshavn, Faroe Islands.

Höskuldur Thráinsson, Hjalmar P. Petersen, Jógvan í Lon Jacobsen, and Zakaris Svabo Hansen. 2012. *Faroese – an overview and reference grammar*, second edition. Faroe University Press and Linguistic Institute, University of Iceland, Tórshavn, Faroe Islands and Reykjavík, Iceland.

Danish Clinical Named Entity Recognition and Relation Extraction

Martin Sundahl Laursen*
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark
msla@mmmi.sdu.dk

Jannik Skyttegaard Pedersen*
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark
jasp@mmmi.sdu.dk

Rasmus Søgaard Hansen
Department of Clinical Biochemistry
Odense University Hospital

Thiusius Rajeeth Savarimuthu
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark

Pernille Just Vinholt
Department of Clinical Biochemistry
Odense University Hospital

Abstract

Electronic health records contain important information regarding the patients' medical history but much of this information is stored in unstructured narrative text. This paper presents the first Danish clinical named entity recognition and relation extraction dataset for extraction of six types of clinical events, six types of attributes, and three types of relations. The dataset contains 11,607 paragraphs from Danish electronic health records containing 54,631 clinical events, 41,954 attributes, and 14,604 relations. We detail the methodology of developing the annotation scheme, and train a transformer-based architecture on the developed dataset with macro F1 performance of 60.05%, 44.85%, and 70.64% for clinical events, attributes, and relations, respectively.

1 Introduction

Electronic health records (EHR) contain important information regarding the patients' medical history including diagnoses, medications, treatment plans, allergies, and test results. However, much of this information is stored in unstructured narrative text. While this information could be used to guide diagnostic decision making and treatment plans, the unstructured format makes it infeasible to fully exploit in clinical practice and research.

Natural language processing (NLP) algorithms could be used to transform the unstructured narrative text of the EHR into structured information and give medical doctors (MD) a fast overview of even a medical history spanning multiple years. NLP models' ability to process and extract information from written text keeps improving with benchmark-breaking models being published on a regular basis. For example, transformer-based models such as GPT-3 (Brown et al., 2020), BERT (Devlin et al., 2019), and ELECTRA (Clark et al., 2020) have recently shown promising results for many NLP tasks, e.g. named entity recognition and relation extraction (NER). In NER, models are trained to tag words with predefined entities and find the relations between them. In clinical NER, entities such as diseases, treatments, drugs, and tests have been extracted automatically from EHRs. However, many of the developed datasets are only in English and for specific clinical specialities or note types (Uzuner et al., 2007, 2010; Bethard et al., 2016).

This paper describes the methodology for developing the first Danish clinical NER dataset. The dataset consists of text paragraphs from Danish EHRs spanning multiple departments and note types.

First, the paper describes the clinical dataset, the strategy for choosing entities tailored to extract important information from EHRs, and the annotation scheme. Next, we train a transformer-based architecture on the developed NER dataset.

2 Related works

The annotation schemes and extracted clinical entities and relations vary. Agrawal et al. (2022) extracted medications, their status (active, discontinued, neither), and attributes. The i2b2 2009 challenge (Uzuner et al., 2010) and n2c2 2018 Track 2 (Henry et al., 2020) only extracted medications

*Equal contribution

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 655–666

and their attributes. Examples of attributes are name, dosage, mode of administration, frequency, duration, reason, strength, form, and adverse drug effects.

SemEval-2016 Task 12 (Bethard et al., 2016) extracted time entities; event entities and their contextual modality, degree, polarity, and type; and temporal relations between time and event entities (before, overlap, before—overlap, after).

SemEval-2015 Task 14 (Elhadad et al., 2015) and CLEF eHealth 2013 Task 1 (Pradhan et al., 2015) extracted disorder mentions and mapped them to their UMLS/SNOMED concept unique identifier. The former also classified attributes such as the disorder's subject, course, body location, and severity, and whether it was negated, uncertain, conditional, or generic.

The i2b2 2010 challenge (Uzuner et al., 2011) extracted entities (medical problems, treatments, tests), assertions (present, absent, possible, conditional, hypothetical future, and associated with someone other than the patient), and relations between medical problem entities and each of medical problem, treatment, and test entities.

The i2b2 2012 challenge (Sun et al., 2013b) extracted clinically relevant events. Their type was classified as concept (problem, test, treatment), clinical department, evidentials indicating source of information, or occurrences (events that happen to the patient). Polarity was classified as positive or negated, and modality as happens, proposed, conditional, or possible. Temporal expressions were extracted with their type (date, time, duration, frequency), value, and modifier indicating whether the temporal expression was exact or not. Temporal relations indicating the type of connections between events and temporal expressions were also extracted.

3 Methods

This section describes the data, annotation scheme, and model used for Danish clinical NER.

3.1 Data

We extracted 11,607 paragraphs with a length between 11 and 75 words from EHRs from Odense University Hospital in Denmark. Paragraphs were sampled randomly from different EHR note types across every department of the hospital to ensure the data distribution would resemble that of EHRs: 46% were from clinical contacts, 13% primary journals, 10% care data, 3% epicrises, 3% ambulatory care contacts, 2% surgical notes, 2% emergency room journals, and 20% were from 55 different minor EHR note types. Paragraphs were lowercased and anonymised by two of the authors.

Clinical event	Description
Disease	A disorder of structure or function, especially one that has a known cause and a distinctive group of symptoms, signs, or anatomical changes. Examples include cancer, influenza, and narcolepsy.
Symptom	A symptom is a physical or mental feature which is regarded as indicating a condition of disease, particularly such a feature that is apparent to the patient. We include abnormal findings, which the MD makes when examining the patient objectively, as these are sometimes coinciding with symptoms—e.g. bruises. Examples include headache, stomach ache, and pain.
Diagnostic	Any tool or method concerned with the diagnosis of illnesses or other problems. Includes measurements and tests. Examples include CT scans, blood samples, and temperatures.
Treatment	A treatment is any medical care given to a patient for an illness or injury. Examples include medication, plaster, and rehabilitation.
Anatomy	Any part of human anatomy. Includes body fluids and excrements. Examples include arms, organs, and blood.
Result	All results of diagnostics that do not carry any meaning without being coupled to the diagnostic. Examples include numbers that indicate length, temperature, or volumes. Diseases or symptoms found by diagnostics are annotated as such, e.g. a tumour found by a CT scan.

Table 1: Description of clinical events. Descriptions were inspired by the Oxford English Dictionary.

3.2 Annotation

3.2.1 Annotation scheme

Two MDs with expert clinical domain knowledge developed the annotation scheme through an iterative process of making annotation rules and testing them.

Annotation rules were made to extract clinically relevant information from the medical history. Focus was for the rules to be as complete as possible to capture all important information about the medical history while still being simple to use for the annotators.

We extracted three types of information: clinical events, the attributes of the clinical events, and relations between the clinical events.

Clinical events were: diseases; symptoms, including abnormal findings; diagnostics; treatments; anatomies including body fluids and excrements; and results. Symptoms and abnormal findings were joined in one as they sometimes coincided. Normal findings were not included as there were so many that they would cloud the visualisation of the history. Table 1 shows all clinical events and their descriptions as defined by the medical experts.

Clinical events were further described by their attributes. Attributes were: prior; current; fu-

Attributes	Description
Prior	Entities that occurred in prior admissions or in the distant past. Includes treatments that are being stopped at that point in time.
Current	Entities that occur in the present. Includes prescribed medicine.
Future	Entities that occur or might occur in the future—e.g. the risk of skin cancer, or ordering diagnostics for a later day.
Doubt	Any entity that is not confirmed. Includes any treatments that might need to be started in the future.
Negation	Entities such as diseases or symptoms that are mentioned as not being present.
Non-patient	Entities that are not related to the patient in question. One example is the disease history of the patient's relatives.

Table 2: Description of attributes.

ture; doubt; negation; and non-patient. All clinical events could take one of the six attributes except anatomies and results. Anatomies did not take any attributes while results could only take a prior or current attribute. Table 2 shows all attributes and their descriptions.

Clinical events could connect to each other in limited ways through one-way relations. Diseases, diagnostics, and symptoms could connect to anatomies through a "has location" relation. Diseases, symptoms, and anatomies could connect to treatments through a "is treated with" relation. Diagnostics could connect to results through a "has result" relation.

Figure 1 shows an overview of the clinical events, attributes, and relations. Appendix A shows the full annotation guidelines with further details and explanations to the annotators.

3.2.2 Annotation process

Six annotators were recruited for the task. Five were Master of Science in Medicine students and one was a MD.

Figure 2 shows the process of annotator training. It included reading the annotation guide and an iterative process of annotating a learning set of 55 paragraphs (not included in dataset) followed by error analysis until a final test was made on a set of 98 gold paragraphs annotated by an expert MD. Paragraphs were annotated using the CLAMP software (Soysal et al., 2017). We report the micro F1 of each annotator on the gold set.

Figure 3 shows an example of an annotated paragraph.

3.3 Entity and relation extraction model

This section describes the architecture of the Princeton University Relation Extraction system (PURE) (Zhong and Chen, 2021) which we used and adapted for Danish clinical NER. It further describes the dataset used and the training of the models.

3.3.1 Model architecture

PURE—the 2021 state-of-the-art on entity and relation extraction—is a NER deep learning model based on a transformer structure. The model has a separate entity and relation extraction part.

For entity extraction, the model takes as input all possible text spans up to a maximum length. A transformer extracts contextual word embeddings for the start and end token of each span. They are concatenated with a learned span width embedding and classified by a feedforward network.

When extracting relations, for each candidate pair of entities, the text is passed through a transformer with inserted entity start and end marker tokens for the subject and object entity, also indicating the type. The concatenation of the start marker token for the candidate subject and object entity is classified by a feedforward neural network.

We used PURE's entity extraction approach for clinical events and the relation extraction approach for relations between clinical events.

We used our own approach adapted from the PURE relation extraction approach for attributes. We inserted clinical event start and end marker tokens, passed all tokens through a transformer, concatenated the start and end marker tokens, and classified the attribute using a feedforward network. The marker tokens were used for classification instead of the word(s) forming the clinical event to guide the model to look more at the context rather than the specific word—the context being the important factor in attribute classification. Additionally, enriching the input with the type of the clinical event could guide the model if attributes were described differently for different clinical events.

Figure 4 shows the three types of extraction tasks.

3.3.2 Datasets

Table 3 shows the number of clinical events, attributes, and relations by type in the train, validation, and test set. The dataset had a total of 11,607 paragraphs, each containing a varying number of clinical events, attributes, and relations. On average, each paragraph contained 4.7 clinical events, 3.6 attributes, and 1.3 relations. We split the paragraphs in train, validation, and test sets for an approximate 80%–10%–10% ratio between each type of clinical event, attribute, and relation. The sets were unbalanced on type of entity or relation—e.g. for the attributes training

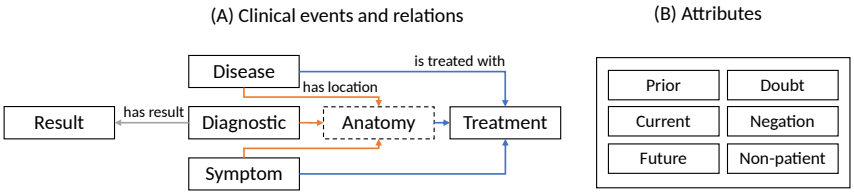


Figure 1: (A) Clinical events and relations between them. Symptoms include abnormal findings. Anatomies include body fluids and excrements. Diagnostics include measurements and tests. Blue: "is treated with". Orange: "has location". Grey: "has result". (B) Attributes. Anatomy (dashed lines) takes no attributes. Other clinical events must take one attribute. Results only take prior or current attributes.

	Train (% of row total)	Validation (% of row total)	Test (%of row total)	Total (% of column total)
Paragraphs	9,687 (83%)	960 (8%)	960 (8%)	**11,607 (100%)**
	Clinical events			
Diseases	2,033 (78%)	295 (11%)	272 (10%)	**2,600 (5%)**
Symptoms	11,937 (80%)	1,455 (10%)	1,571 (10%)	**14,963 (27%)**
Diagnostics	8,921 (80%)	1,095 (10%)	1,194 (11%)	**11,210 (21%)**
Treatments	6,918 (79%)	911 (10%)	882 (10%)	**8,711 (16%)**
Anatomies	10,172 (80%)	1,227 (10%)	1,278 (10%)	**12,677 (23%)**
Results	3,522 (79%)	473 (11%)	475 (11%)	**4,470 (8%)**
TOTAL	**43,503 (80%)**	**5,456 (10%)**	**5,672 (10%)**	**54,631 (100%)**
	Attributes			
Prior	2,028 (80%)	237 (9%)	283 (11%)	**2,548 (6%)**
Current	23,217 (79%)	3,021 (10%)	3,109 (11%)	**29,347 (70%)**
Future	1,237 (79%)	161 (10%)	160 (10%)	**1,558 (4%)**
Doubt	2,479 (82%)	263 (9%)	289 (10%)	**3,031 (7%)**
Negation	3,890 (80%)	496 (10%)	500 (10%)	**4,886 (12%)**
Non-patient	480 (82%)	51 (9%)	53 (9%)	**584 (1%)**
TOTAL	**33,331 (79%)**	**4,229 (10%)**	**4,394 (10%)**	**41,954 (100%)**
	Relations			
is treated with	1,485 (80%)	175 (9%)	197 (11%)	**1,857 (13%)**
has location	6,501 (80%)	779 (10%)	823 (10%)	**8,103 (55%)**
has result	3,652 (79%)	499 (11%)	493 (11%)	**4,644 (32%)**
TOTAL	**11,638 (80%)**	**1,453 (10%)**	**1,513 (10%)**	**14,604 (100%)**

Table 3: Composition of the train, validation and test sets by type of clinical event, attribute, and relation.

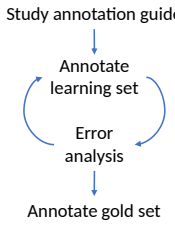


Figure 2: Annotator training process. Figure inspired by Sun et al. (2013a).

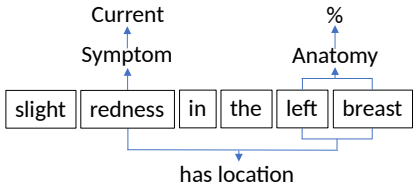


Figure 3: Example of annotated paragraph. % signifies that no attribute could be assigned to the clinical event per the annotation scheme.

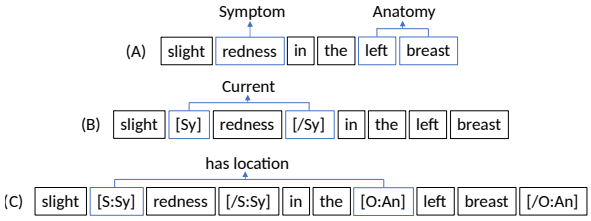


Figure 4: (A) Classification of clinical events from start and end tokens of span. Span width embedding not depicted. (B) Classification of attribute using clinical event marker tokens. (C) Classification of relation using subject/object and clinical event marker tokens. Figure inspired by Zhong and Chen (2021).

Evaluation metric	Loss	Micro			Macro		
		R%	P%	F1%	R%	P%	F1%
Micro F1	**Unweighted**	79.14	79.14	79.14	38.34	40.51	38.56
	Weighted	61.81	61.81	61.81	45.35	33.20	34.23
Macro F1	**Unweighted**	77.30	77.30	77.30	41.88	41.90	41.48
	Weighted	60.13	60.13	60.13	51.37	41.87	43.85

Table 4: Validation set micro and macro recall, precision, and F1 score on the attribute extraction task when selecting the best iteration of the model based on micro and macro F1 score with unweighted and weighted loss. 2 hidden layers of size 75 was used for the test. R: Recall. P: Precision.

set, there were 23,217 current and only 480 non-patient attributes. All datasets were in the json format used by PURE (see Zhong and Chen (2021)).

3.3.3 Training

When training the clinical event extraction model, we used a Danish Clinical ELECTRA pretrained on the narrative text from 299,718 EHRs from Odense University Hospital as the transformer base (Pedersen et al., 2022). The model had ∼13M parameters and consisted of 12 transformer layers with 4 attention heads. We used a dropout of 0.1 after the last ELECTRA hidden layer output. We tested classification heads with two hidden layers of varying size, each followed by a dropout of 0.2 and a ReLU activation function. We used a maximum span of 8 and a train batch size of 32. We trained for 100 epochs using the AdamW optimizer with learning rate 1e-5 for the transformer layers and 1e-4 for the classification head, and a warm-up proportion of 0.1.

When training each of the models for extracting attributes and relations, we used the same transformer base with a normalisation layer and a dropout of 0.1 after the concatenation of tokens. We tested classification heads with two hidden layers of varying size, each followed by a dropout of 0.2 and a ReLU activation function. We further tested a classification head only consisting of a single classification layer. We used a train batch size of 32 and a maximum sequence length of 128. We trained for 20 epochs using the AdamW optimizer with learning rate 2e-5 and a warm-up proportion of 0.1.

We modified the training method of PURE to guide the models towards equal performance on all classes by using a weighted loss function to counteract the unbalanced dataset and chosing the best model for each of the clinical event, attribute,

and relation extraction tasks as the model iteration with the best macro F1 on the validation set, rather than the micro F1 standard of PURE. Table 4 shows a test of the performance on the attribute extraction task when selecting the best iteration of the model based on micro and macro F1 score with unweighted and weighted loss. Using the macro F1 score with weighted loss gave the best performance across all classes. Appendix B shows the confusion matrices for each combination.

Class weights were calculated for the training of each model using the default formula in Scikit-learn (Pedregosa et al., 2011):

$$w_x = \frac{n_{samples}}{n_{classes} \cdot n_x} \quad (1)$$

where x is the class, $n_{samples}$ is the number of total samples, and $n_{classes}$ is the number of classes. The negative class, i.e. samples not to be given any label by the model, was given a weight of 1.

The negative class was excluded when calculating the F1. We only trained the attribute and relation models to make classifications that were allowed for the connected clinical events according to the annotation scheme. Appendix C shows the results of the hyperparameter search. We report the micro and macro recall, precision, and F1 for the best models on the test set.

4 Results

This section presents the agreement of the annotators on the gold set and the results of the Danish clinical NER models.

4.1 Annotation

Table 5 shows the annotators' micro F1 performance on the gold set. For clinical events, it ranged 83.71%–91.24% (average 85.62%) for overlapping matches, and 74.12%–85.15% (average 77.67%) for exact matches. For attributes, it ranged 79.21%–86.19% (average 81.71%) and for relations 71.28%–90.06% (average 77.79%).

4.2 Entity and relation extraction model

The models that had the best validation performance in the hyperparameter search were:

- A clinical event extraction model with two hidden layers of size 450 in the classification head.
- An attribute extraction model with a single classification layer.
- A relation extraction model with two hidden layers of size 150 in the classification head.

Annotator	A	B	C	D	E	F
	Overlap match, micro F1%					
Clinical event	91.24	84.22	84.41	85.71	84.43	83.71
Attribute	86.19	83.06	79.21	81.29	79.75	80.75
Relation	90.06	76.97	75.60	77.01	71.28	75.84
	Exact match, micro F1%					
Clinical event	85.15	76.08	76.29	78.69	74.12	75.71

Table 5: The anonymised annotators' performance on the gold set. Exact match: a match is defined as the exact tokens annotated in the gold set with the same label. Overlap match: a match is defined as minimum one token overlapping with the gold set annotation of the same label. Only an overlap match F1 is calculated for attributes and relations as evaluating an exact match would propagate the potential error in the span of the clinical event to which the attribute or relation is connected.

	Micro			Macro		
	R%	**P%**	**F1%**	**R%**	**P%**	**F1%**
	Overlap match					
Clinical events	66.29	77.31	71.38	64.88	72.60	68.20
	Exact match					
Clinical events	60.97	65.64	63.22	59.84	61.30	60.05
Attributes	66.04	66.04	66.04	51.60	42.64	44.85
Relations	75.88	72.66	74.23	74.74	67.85	70.64

Table 6: Performance of the best clinical event, attribute, and relation extraction models on the test set. Attributes and relations are only reported with an exact match as the models do not consider the span of the clinical event from which the attribute or relation is classified. R: Recall. P: Precision.

Table 6 shows the performance of the best models on the test set. Clinical events were extracted with exact micro F1 63.22% and macro F1 60.05%, attributes with micro F1 66.04% and macro F1 44.85%, and relations with micro F1 74.23% and macro F1 70.64%. The negative class was excluded when calculating the recall, precision, and F1 scores.

Figure 5 shows the confusion matrices of performance on clinical events, attributes, and relations. The confusion matrices include the clinical events and relations that were not extracted and falsely extracted by the model ('O').

The model for clinical event extraction performed best on anatomies (69%) and worst on results (53%). 1,568 spans were falsely extracted

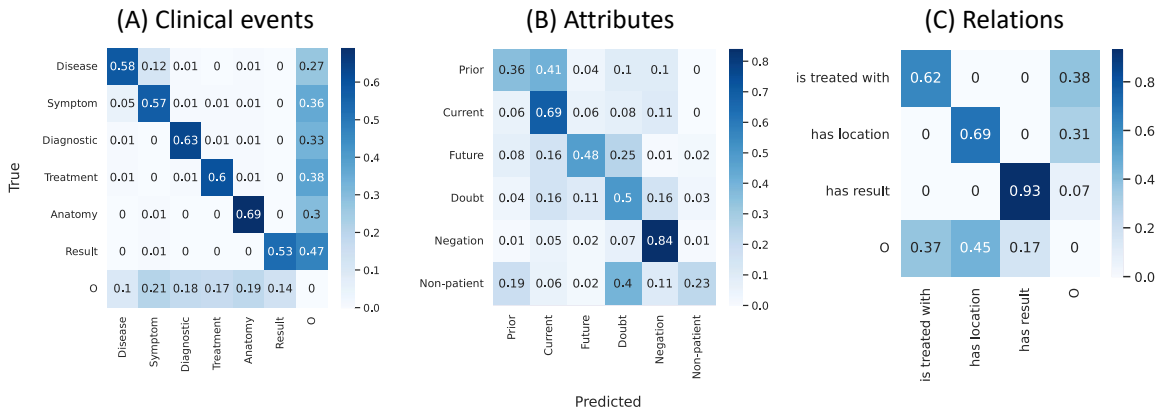


Figure 5: Confusion matrices of performance on (A) clinical events, (B) attributes, and (C) relations. 'O' counts the clinical events and relations that were not extracted and falsely extracted by the model.

as a clinical event with symptoms being the most frequent (21%). The model for attribute extraction performed best on negations (84%) and worst on non-patient (23%). The model for relation extraction performed best on "has result" (93%) and worst on "is treated with" (62%). 432 false relations were extracted of which "has location" was the most frequent misclassification (45%).

5 Discussion and limitations

This paper presented a methodology for developing a dataset for Danish clinical NER. It presented an annotation scheme for annotation of all clinical events, their attributes, and relations that are relevant for the medical history. The dataset included text paragraphs from Danish EHRs spanning multiple departments and note types.

We trained and adapted PURE NER deep learning models to extract clinical events (overlap match macro F1 68.20%; exact match macro F1 60.05%), attributes of clinical events (macro F1 44.85%), and relations between clinical events (macro F1 70.64%). The results are promising for Danish clinical NER but need improvement. A discussion of possible improvements to the methodology, limitations, and future work is provided below.

The clinical event extraction model had similar performance on all classes with accuracies between 53% (results) and 69% (anatomies). There was little contamination between classes as most errors were caused by failure to extract or false extraction of a clinical event. There was some contamination between symptoms and diseases with 12% of diseases being classified as symptoms and 5% of symptoms being classified as diseases. This supports claims by annotators that diseases and symptoms in some cases are difficult to differentiate and that extra attention must be given to differentiate these in the annotation guidelines.

The attribute extraction model had large differences in performance with accuracies between 23% (non-patient) and 84% (negation). There were more misclassifications of the non-patient attribute as doubt (40%) than correct classifications. The future and doubt attributes had significant contamination between them with 25% and 11% misclassifications as the other class, respectively. The many misclassifications between non-patient and doubt attributes, and especially future and doubt attributes, could indicate that the model would improve if the non-patient, doubt, and future attributes were merged to a single class of uncertain attributes. This would most likely not harm the usefulness of the model to MDs significantly.

The fact that more prior attributes were misclassified as current (41%) than correct classifications (36%) likewise indicates that these two attributes could be merged into a single class of clinical events that occurred. This would, however, decrease the usefulness of the model as it is important for MDs reviewing the medical history to know if a clinical event is prior or current.

The relation model extracted 93% of the "has result" relations, and 62% and 69% of the "is treated with" and "has location" relations, respectively. The differences are likely caused by the fact that the "has result" relation only connects diagnostics to results while the two other relations have three different one-way relationships.

In this paper, we only explored one type of NER model and tested a limited set of architectures and hyperparameters. Future work could include testing other architectures and enriching the model input with more information, e.g. the output of a text parser, which could help differentiate attributes dealing with the time-aspect. The six annotators had an average micro F1 (overlap match) of 85.62%, 81.71%, and 77.79% for clinical events, attributes, and relations, respectively. Merging certain attributes and more emphasis on differences between symptoms and diseases could increase these scores.

The Danish clinical NER dataset is not made publicly available due to it containing sensitive information. We advise interested researchers to contact us for sharing possibilities.

6 Conclusions

This paper presented methodology and annotation scheme for developing the first Danish clinical NER dataset. The corpus consists of 11,607 paragraphs annotated for six entity types, six attributes, and three relations. The corpus was used to fine-tune language models which showed promising results for classifying the entities, attributes, and relations of the dataset.

References

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. 2022. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022.

Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. Semeval-2016 task 12: Clinical tempeval. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 1052–1062.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. https://openreview.net/forum?id=r1xMH1BtvB Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. https://doi.org/10.18653/v1/N19-1423 BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Noémie Elhadad, Sameer Pradhan, Sharon Gorman, Suresh Manandhar, Wendy Chapman, and Guergana Savova. 2015. Semeval-2015 task 14: Analysis of clinical text. In *proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 303–310.

Sam Henry, Kevin Buchan, Michele Filannino, Amber Stubbs, and Ozlem Uzuner. 2020. 2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records. *Journal of the American Medical Informatics Association*, 27(1):3–12.

Jannik S Pedersen, Martin S Laursen, Cristina Soguero-Ruiz, Thiusius R Savarimuthu, Rasmus Søgaard Hansen, and Pernille J Vinholt. 2022. Domain over size: Clinical electra surpasses general bert for bleeding site classification in the free text of electronic health records. In *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 1–4. IEEE.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Sameer Pradhan, Noémie Elhadad, Brett R South, David Martinez, Lee Christensen, Amy Vogel, Hanna Suominen, Wendy W Chapman, and Guergana Savova. 2015. Evaluating the state of the art in disorder recognition and normalization of the clinical narrative. *Journal of the American Medical Informatics Association*, 22(1):143–154.

Ergin Soysal, Jingqi Wang, Min Jiang, Yonghui Wu, Serguei Pakhomov, Hongfang Liu, and Hua Xu. 2017. https://doi.org/10.1093/jamia/ocx132 CLAMP – a toolkit for efficiently building customized clinical natural language processing pipelines. *Journal of the American Medical Informatics Association*, 25(3):331–336.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013a. https://doi.org/https://doi.org/10.1016/j.jbi.2013.07.004 Annotating temporal information in clinical narratives. *Journal of Biomedical Informatics*, 46:S5–S12. Supplement: 2012 i2b2 NLP Challenge on Temporal Relations in Clinical Data.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013b. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813.

Özlem Uzuner, Yuan Luo, and Peter Szolovits. 2007. Evaluating the state-of-the-art in automatic de-identification. *Journal of the American Medical Informatics Association*, 14(5):550–563.

Özlem Uzuner, Imre Solti, and Eithon Cadag. 2010. Extracting medication information from clinical text. *Journal of the American Medical Informatics Association*, 17(5):514–518.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61.

Appendices

A Annotation guidelines

A.1 Clinical events

A.1.1 Disease

Contains all diseases including diseases that could be considered a result of a Diagnostic.

A.1.2 Symptom

Includes all symptoms and abnormal findings. Findings that are not abnormal should not be annotated. However, a negation of an abnormal finding should be annotated because the abnormal finding is mentioned even though it is not present. For example, "fracture" should be annotated in the sentence "there is no sign of fracture."

If there is a negation of a non-abnormal finding, it should be annotated in the entity. For example, "cannot hear" is annotated in the sentence "patient cannot hear anything."

In the sentence "no symptoms," the word "symptoms" should not be annotated as a symptom, as it does not contain any information.

In case a symptom or abnormal finding is found by a Diagnostic, there may be a coincidence with the Result entity. Here, it is annotated as Symptom if the entity can provide sufficient meaning alone. For example, "cyst" or "tumour."

If the Symptom cannot stand alone and one needs to know which Diagnostic was carried out in order to understand the result, the entity should instead be annotated as Result and have a "has result" relationship from the Diagnostic entity. For example, this applies to "Temp: 24 C" and "Stix: 3+". "Temp" and "Stix" are annotated as Diagnostic with "is treated with" relationship to Result "24 C" and "3+."

A.1.3 Result

Includes all results of Diagnostic, e.g. values and blood test results.

A Result cannot stand on its own. A relation from the Diagnostic is needed for it to make sense. These can be entities like "stable", "positive", "negative", "24 C" or "3+".

Typically, this entity will appear in sentence structures with a colon: "Diagnostic: Result". Note that the two entities are mentioned very close to each other in the text—in this case only with a colon in between. An example could be "Temp: 24 C" or "Stix: 3+". "Temp" and "Stix" are annotated as Diagnostics with a "has result" relation to Result "24 C" and "3+".

Entities that can instead be annotated as Symptom will typically be mentioned further away or completely lack a Diagnostic as a Symptom can stand alone and make sense.

See also the description for Symptom.

A.1.4 Diagnostic

Includes all diagnostics, measurements, and tests. This can include CT scans, blood tests, MR scans, and recordings of a newborn's length, temperature, etc.

Note that "blood sample results" and "radiology description" are not a Diagnostic and should not be annotated.

If KAD is mentioned along with a volume, e.g. "KAD emptied of 200 mL," it is marked as Diagnostic - Result. If there is no volume specified, KAD is annotated as Treatment.

A.1.5 Treatment

Includes all forms of treatment including medication.

To annotate entities as concisely as possible, for example in the sentence "good effect of 2.5 mg morphine IV," only "morphine" should be annotated as Treatment.

In the sentence "treated for xxx," the word "treatment" should not be annotated as Treatment as it does not contain any information.

If KAD is mentioned without a volume indication, it should be annotated as Treatment. If KAD is mentioned with a volume, for example "KAD emptied for 200 mL," it should be annotated as "Diagnostic - Result."

A.1.6 Anatomy

Includes all mentions of anatomies and things from the body (blood, feces, urine, sweat, etc.).

Typically used to indicate the location of a Disease or Symptom, a Diagnostic, or a Treatment. Examples: "brain", "left foot" or "duodenum".

When Anatomy is described by an adjacent word, for example "left", this should be included in the entity.

Remember to annotate the Anatomy entities that should not be linked to other entities.

A.2 Attributes

A.2.1 Current

The entity is either present, carried out, or current. If medication is prescribed to the patient, this should also be marked as "Treatment - Current", as it can be assumed that the treatment will start and it may be the last time it is mentioned in the journal. On the other hand, "Scheduling a CT for Tuesday." should be marked as "Future" as it will be described in a future medical note, for example with the result.

A.2.2 Negation

The entity is not present. For example, if it is mentioned that the patient does not have a fracture, the fracture should be marked as Symptom - Negation. Note that the word "not" should not be part of the marked entity. However, if there is a negation of a normal finding, it should be annotated as such. For example, "cannot hear" in the sentence "patient cannot hear anything" is annotated as Symptom - Present.

A.2.3 Prior

If the entity refers to a previous case, i.e., a previous hospitalisation or if it happened a long time ago. For example, it should be annotated as a prior Treatment when a cast or drain is removed, as the treatment is finished. However, if a CT scan from the previous day is mentioned, it should be annotated as Current.

A.2.4 Future

Everything that takes place in the future. For example, cancer is annotated as Disease - Future if it is mentioned that "there is a risk of cancer if you use tanning beds too often."

It is marked as Diagnostic - Future if an MRI scan is planned for the next day. However, if it is written "the treatment with xxx starts" or "rp. xxx" it should be marked as Treatment - Current as it is assumed that the treatment will certainly happen.

Also includes references to possible future treatments.

A.2.5 Doubt

If the patient might have a disease that has not yet been confirmed.

If a Treatment should be given provided that certain things change.

The difference between Doubt and Future is that Future is more certain - it is going to happen - while Doubt is more uncertain or conditional.

A.2.6 Non-patient

If an entity does not have a direct connection to the patient. This can occur when a general letter is sent out regarding cancer screening. Cancer should then be annotated as Disease - Non-patient. If it is mentioned that the patient's mother had a certain disease, it should also be annotated in this way.

A.3 Relations

When entities are annotated, the relationships between entities can be annotated. This is done by pulling the "From entity" over to the "To entity". The direction of the relationship is important. Therefore, pay attention to the name of the relationship and read it out loud if necessary, "Entity - Relation - Entity" and listen to see if it makes sense or if the arrow needs to be reversed. CLAMP will show which relationships can be annotated for the pair being drawn between.

has location

From entities: Disease, Symptom, Diagnostic.
To entities: Anatomy.

has result

From entities: Diagnostic.
To entities: Result.

is treated with

From entities: Disease, Symptom, Anatomy.

To entities: Treatment.

The "is treated with" relation links the entities Disease, Symptom, and Anatomy to a Treatment. In some cases, sentences describing a required treatment could be linked to both an Anatomy and Treatment entity. In this case, the Treatment should be linked to the Symptom instead of the Anatomy. You should only link the Anatomy to the Treatment using the "is treated with" relation if the Treatment cannot be linked to anything else. Example: "Left knee skin scraping is treated with plaster." Annotation: skin scraping - "Treated with" - plaster.

A.4 General notes

It is important not to annotate periods, commas, etc. unless they are part of an abbreviation. For example, in "Patient has cancer," only "cancer" and not "cancer." should be marked. If you double-click a word, CLAMP will only mark the word and not any punctuation next to the word. This can make it a bit troublesome to include periods in abbreviations.

Entities should be annotated as concisely as possible without losing meaning. This means that in the sentence "there are signs of cancer," only "cancer" and not "signs of cancer" should be marked as an entity. If an entity has some describing words next to it, the following rule can be used to decide how much should be annotated. In the sentence "pain in the front of the arm," only "arm" is marked as Anatomy since "front" and "arm" are connected through the word "of." In the sentence "pain in the left arm," "left arm" is marked as Anatomy since there are no words between "left" and "arm". In sentences describing a prescription of medication, only the name is marked as Treatment, and not, for example, the quantity indication or the number of days.

Entities may not overlap with each other.

B Selection of loss and evaluation metric

Figure 6 shows the confusion matrices for the attribute extraction task when selecting the best iteration of the model based on micro and macro F1 score with unweighted and weighted loss.

Using the micro F1 to select the best iteration of the model resulted in some classes being practically excluded during classification. Using the macro F1 to select the best model iteration and training with a weighted loss gave the most equal performance on all classes.

	Classification head hidden layers	Validation Exact F1 %
Clinical event	2x 75	58.49
	2x 150	59.82
	2x 300	60.68
	2x 450	61.34
	2x 600	60.91
Attribute	None	48.01
	2x 50	43.20
	2x 75	43.85
	2x 150	44.10
	2x 300	44.32
Relation	None	66.15
	2x 75	68.39
	2x 150	68.85
	2x 300	67.39

Table 7: Results of the hyperparameter search.

C Hyperparameter search

Table 7 shows the results of the hyperparameter search.

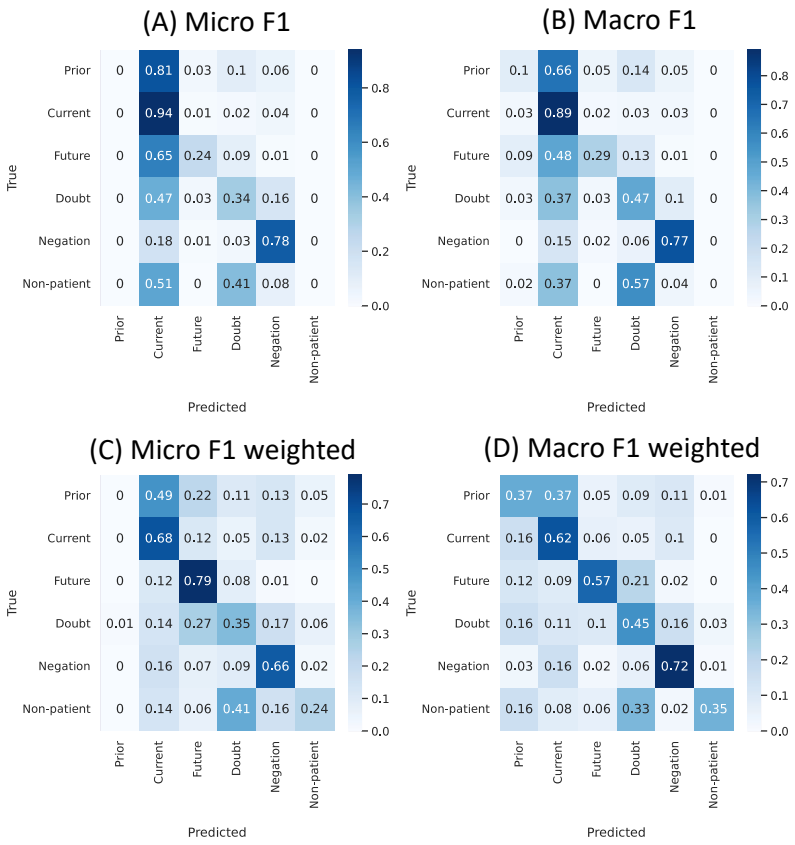


Figure 6: Confusion matrices showing the attribute extraction validation performance of the models chosen based on (A) micro F1, (B) macro F1, (C) micro F1 trained with weighted loss, and (D) macro F1 trained with weighted loss.

Scaling-up the Resources for a Freely Available Swedish VADER (*svVADER*)

Dimitrios Kokkinakis
University of Gothenburg and Centre for Ageing and Health (AgeCap)
dimitrios.kokkinakis@gu.se

Ricardo Muñoz Sánchez
University of Gothenburg
ricardo.munoz.sanchez@gu.se

Mia-Marie Hammarlin
Lund University and Birgit Rausing Centre for Medical Humanities (BRCMH)
mia-marie.hammarlin@kom.lu.se

Abstract

With widespread commercial applications in various domains, sentiment analysis has become a success story for Natural Language Processing (NLP). Still, although sentiment analysis has rapidly progressed during the last years, mainly due to the application of modern AI technologies, many approaches apply knowledge-based strategies, such as lexicon-based, to the task. This is particularly true for analyzing short social media content, e.g., tweets. Moreover, lexicon-based sentiment analysis approaches are usually preferred over learning-based methods when training data is unavailable or insufficient. Therefore, our main goal is to scale-up and apply a lexicon-based approach which can be used as a baseline to Swedish sentiment analysis. All scaled-up resources are made available, while the performance of this enhanced tool is evaluated on two short datasets, achieving adequate results.

1 Introduction

Sentiment analysis is the computational study of people's opinions, sentiments, emotions, and attitudes towards entities such as products and services, and their attributes. Sentiment analysis allows tracking of the public's mood about a particular entity to create actionable knowledge (Ligthart et al., 2021) and has found numerous applications, ranging from digital humanities (Kim & Klinger, 2022) to gaining insight into customers' feedback about commercial products and services (Rashid & Huang, 2021). Sentiment analysis can occur at the document, sentence, or word level, while the sentiment types usually assigned are *Very positive*, *Positive*, *Neutral*, *Negative* or *Very negative*. E.g., the sentiment for the sentence *Att känna stödet från publiken och folket är väldigt smickrande* 'To feel the support of the audience and the people is very flattering' will be usually assigned a positive sentiment while the sentence *Föräldrar i chock efter bluffen i basketlaget* 'Parents in shock after the hoax in the basketball team' will be assigned a negative one.

In this paper we discuss an enhancement of a popular off-the-shelf (unsupervised) dictionary-based approach to Sentiment analysis using VADER (*Valence Aware Dictionary and sEntiment Reasoner*). VADER is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media (Hutto & Gilbert, 2014). VADER is fully open-sourced, available e.g., from the NLTK package (Bird et al., 2009), which can be applied directly to unlabeled text data. Furthermore, VADER can efficiently handle large vocabularies, including the use of degree modifiers and emoticons. These qualities make VADER a good fit for use on social media input for rapid sentiment text analysis. As such, the need for previous training as in machine or deep learning models, is eliminated.

Our main aim of this work is to make VADER a useful baseline for Swedish sentiment analysis, by rapidly scaling-up and improving the coverage of the already translated to Swedish resources (lexicons and processing tools). We further evaluate the coverage by applying and comparing the original VADER translation with the enhanced version on two small datasets, one with Swedish tweets and one sample from the *ABSAbank-Imm*[1], an annotated Swedish corpus for aspect-based se-

[1] https://spraakbanken.gu.se/en/resources/absabank-imm.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 667–672

ntiment analysis (Rouces et al., 2020).

2 VADER

The *Valence Aware Dictionary for sEntiment Reasoning* (VADER [2]) is a parsimonious rule-based model for sentiment analysis of specifically social media text (Hutto & Gilbert, 2014). Since its release VADER has been extensively used in various applications and domains; from the analysis stock news headlines (Nemes & Kiss (2021); to the assessment of sentiments expressed in customers' e-mails (Borg and Boldt, 2020); and further to the analysis of tweets on COVID-19 vaccine hesitancy (Verma et al., 2022).

2.1 VADER translations

VADER lexical components have been translated into several languages, such as German, French, and Italian [3]. The Swedish translation of the VADER sentiment lexicon, along with the VADER application's negators and booster words, were translated from English to Swedish, using the Google Cloud Translation API by Gustafsson (2019). However, one third of the original English sentiment lexicon remained untranslated during this process, which in a sense decrease the quality of the analysis. According to Gustafsson (2019) the original English VADER lexicon contained 7517 words, slang words, abbreviations, and emoticons. Out of these, 2435 could not be translated to Swedish because no translation could be found; for instance, many words in the English lexicon had inflections that did not exist in the Swedish counterpart; polysemy created problems, as well as English idiomaticity, e.g., slang words remained untranslated. This version of the Swedish VADER sentiment resources can be found in Github[4].

2.2 Enhancements of the translated Swedish VADER: *single words, lexicalized idioms, and other multiword expressions*

The original Swedish translation of VADER was the starting point for developing and enhanced version of VADER (svVADER [5]). In general, VADER is based on a few key points when determining the sentiment of a text:

- *degree modifiers or booster words*, that is dampeners and intensifiers, i.e., words or characters that affects the magnitude of the polarity by either increasing or decreasing the intensity of the sentiment;
- *negations,* words which reverse the semantic orientation in a text and thus also its polarity score;
- *capitalization*, which increases the intensity of polarity, and the sentiment becomes intensified, and,
- certain types of *punctuation*, specifically exclamation marks which increase the intensity of polarity without affecting the semantic feeling.

We started refining and adapting the VADER script, in which booster words and negation items are hard coded. We both added new booster lexical items (e.g., *knappast*; *minimalt*; *svagt*; and *måttligt*) and deleted several dubious words (e.g., *effing*; *flippin*; *frackin*; *fuggin* and *hella*); similarly, some missing Swedish negation words (e.g., *icke*; *inget*; *inga* and *ej*) were also added to this script.

The characterization of the multiword expressions (MWE) and their idiomaticity play an important role in lexically based sentiment analysis. For instance, Moreno-Ortiz et al. (2013) discuss that MWEs, being units of meaning, their relative weight to the calculated overall sentiment rating of texts needs to be accounted for as such, rather than the number of component lexical units. Therefore, we added a list of 100 sentiment laden idioms, that is multiword expressions the meaning of which cannot be deduced from the literal meaning of constituent words (e.g., the Swedish idioms *blåst på konfekten* 'to be cheated on' and the Swedish idiom *tomtar på loftet* which is used to refer to someone who is stupid or crazy). The lexicalized idioms originate from the available list of the NEO lexicon DB [6] that contains a large number (over 4,000) of lexicalized idioms; the selection was made by matching all items on Tweeter and Flashback corpora, extracting the matches, and browsing manually the matched idioms annotating relevant items as positive or negative. Moreover, we manually annotated and

[2] github.com/cjhutto/vaderSentiment.

[3] See here German (Tyman et al., 2019) (github.com/KarstenAMF/GerVADER); French (github.com/vr0nsky/vadersentiment_fr) and here details for Italian (Martinis et al., 2022).

[4] github.com/AlexGustafsson/vaderSentiment-swedish.

[5] github.com/XdimitrisX/svVADER.

[6] spraakbanken.gu.se/en/resources/neo-idiom.

added over 200 phrasal verbs, (e.g., *spränga ihjäl sig* 'to blow yourself up'; *skälla ut* 'to scold'; *rusta ner* 'to gear down' and *rusta upp* 'to gear up'). Statistically significant collocations were also added, these were extracted from the analysis of the two larger collections where the two datasets originate from (cf. Section 3). Also, common medical terminology[7] (i.e., roughly 500 symptoms and frequent disease names) where added with negative polarity to svVADER's main lexicon. Finally, we created an emoj[8] list (3,500) with Swedish expansion (meaning) downloaded and refined from various Internet sites, i.e., 😂 *ansikte med glädjetårar* 'face with tears of joy'.

Table 1 shows the current lexical content of the original and enhanced versions of svVADER.

Name	Size	License
Original translation	5,501	MIT License
Enhanced *single words*	58,070	CC BY 4.0*
Enhanced *MWE*	2,300	

Table 1: The size of the Swedish lexicons (single words: includes inflected forms; MWE: Multi-Word Expressions; '*': license of the enhanced lexicons).

3 Application scenario: Swedish tweets about mRNA vaccines and Flashback posts on immigration

As an application scenario for the evaluation of svVADER we selected two small datasets. The first one consists of Swedish tweets posted in 2022 that discuss vaccine skepticism, and particularly, anxiety about possible side effects and concerns related to novel vaccine technologies, such as the messenger RNA (mRNA) which has be used as a reason for not receiving (the COVID-19) vaccine (Leong et al., 2022). The extracted Swedish tweets were collected with the keywords m-?RNA.* ('?' the preceding character is optional.; '.*': ≥ 0 characters) or the hashtag #mRNA and lang:sv (Swedish content). From the extracted tweets (ca 1,800), a random selection of 200 tweets was selected for the svVADER evaluation. The second dataset originates from the ABSAbank-Imm (where ABSA stands for "Aspect-Based Sentiment Analysis" and Imm for "Immigration", a subset of the Swedish ABSAbank) annotated dataset (Rouces et al., 2020) where we randomly extracted 315 posts. ABSA models predict the sentiment of specific aspects present in the text, that is sentiment expressions that contain no polarity markers but still convey clear human-aware sentiment polarity in context (Russo et al., 2015). In ABSAbank-Imm, texts and paragraphs are manually labelled according to the sentiment (on 1-5 scale) that the author expresses towards immigration in Sweden (a task also known as stance analysis). The 315 posts come from the Flashback Forum[9], a popular Swedish discussion platform. For simplicity, the extracted posts consisted of posts with 1-2 sentences; posts that consisted of 3 or more sentences were excluded. Moreover, the selected posts were labelled as positive if their manually assigned score in ABSA was 5.0 (very positive) or 4.0 (positive) and negative if their manually assigned score was 1.0 (very negative) or 2.0 (negative). Posts that lied in the middle scale with ratio 3.0 were labelled as neutral. Thus, for practical reasons, we collapsed the scores 5.0 and 4.0 to positive sentiment and 1.0 and 2.0 to negative.

3.1 Experimental results and evaluation

The ABSAbank-Imm dataset was already manually labelled, while the Tweeter dataset was manually labelled by one of the authors and a Master student, the inter-annotator agreement[10] was high (Fleiss' $\kappa \approx 0.839$).

VADER's sentiment score is returned in both as a compound score or as *pos*itive, *neg*ative, and *neu*tral. The compound score is computed by summing the valence scores of each word in the text, adjusted according to the rules, and then normalized to be between –1 (very negative) and +1 (very positive). Specifically, VADER's compound sentiment score determines the underlying sentiment of a text (i.e., tweet or post) according to the following schema:

- positive, compound score ≥ 0.05
- negative, compound score ≤ – 0.05
- a neutral, the compound score is between > – 0.05 and < 0.05

We use the *original* Swedish VADER translation to automatically classify each tweet and each Flashback post according to its semantic

[7]Motivated by the fact that there is a growing interest to analyzed social media with health-related content.
[8]https://emojipedia.org/sv/.
[9]https://www.flashback.org/.
[10]For the interrater reliability and agreement, we applied the R package *irr* 0.84.1.

orientation and we then proceed to classify the same data with the enhanced resources. Table 2 summarizes the results of the evaluation, which clearly shows, as expected, that the enhanced approach improved the compound score results based on the original VADER translation.

Modell	CS: Tweets	F1 ABSA
VADER	36,7%	37,2%
svVADER$_{single\ words}$	50,8%	48,2%
svVADER$_{all}$	51%	48,1%

Table 2: Evaluation results for the original Swedish translation of VADER and the enhanced flavors of svVADER. (CS: Compound Score; svVADER$_{single\ words}$: original plus *new* non-MWE words).

For the evaluation of (sv)VADER's performance, we apply a slightly adopted version of the SemEval-2017 Task 4 (Rosenthal et al., 2017), evaluation script[11]. As with other approaches to sentiment analysis there are several pros and cons to the task. The approach is relatively easy to implement and understand, and, given the magnitude of customer experience for products and services available online it becomes doable to capture relevant datasets. However, since the model is primarily designed for use with social media content in mind, the analysis may easily overlook important words or usage. Social media input is usually loaded with typos, misspellings, slang, and grammatical mistakes, including the misinterpretation of ironic or sarcastic statements. Moreover, (sv)VADER ignores the context of the words it analyzes, particularly when word order and discontinuous structures involve cases where the insertion of e.g., one or more lexical items, appears between a lexicalized multiword entry and at a longer distance than the very near context.

4 Conclusions and future work

VADER offers a simple process for sentiment classification with a design focus on social media texts, where no training data is required, and can be used as a baseline method to evaluate and compare other methods. In this paper we outlined the scaling-up process for a dictionary approach to Swedish sentiment analysis using the VADER, a less resource-consuming lexicon and rule-based sentiment analysis tool that consumes fewer resources as compared to learning models as there is no need for vast amounts of training data. As such, VADER can serve as a good starting point to sentiment analysis before diving into more advanced machine learning (e.g., transfer learning; Prottasha et al., 2022); semiautomatic lexicon based (Chanlekha et al., 2018; Barriere & Balahur, 2020) or deep learning models[12] which stand out in terms of usage the last years, and compare their results (Dang et al., 2020). For higher level of accuracy, it may be worth evaluating alternatives (Farah & Kakisim, 2023) or even better a combination of alternative models using the VADER's sentiment scores as input feature to ensemble learning (Kazmaier & vanVuuren, 2022).

The performance of svVADER was further evaluated on two, rather small, but characteristic Swedish social media datasets. One that contains 200 tweets and one with 200 single-sentenced posts from Flashback and the achieved results were adequate. E.g., compared to GerVADER: F1-score=39,42% on German human labelled Tweets and VADER-IT, Gynaecology reviews, with F1-score=50.47%. We have also shown several ways to augment and expand the resources, and there is a strong indication in which MWEs can slightly contribute to the improvement of the results (semantic orientation) of the texts. Perhaps evaluation on much larger and varied datasets could achieve better performance.

Limitations

There are many challenges with this approach. The representativity of the tweets or the social media sample, and their size is low and polarized, further experimentation is necessary on larger, manually curated datasets to verify the efficacy of the tool and resources on different domains and text genres. Apart from the text selection process, this paper didn't provide a comparison with learning methods, a task we left for future research.

[11] https://github.com/cardiffnlp/xlm-t/blob/main/src/evaluation_script.py

[12] A starting point could be the Swedish BERT models for sentiment analysis: Recorded Future & AI Sweden https://huggingface.co/RecordedFuture/Swedish-Sentiment-Fear. The two models are based on the KB/bert-base-swedish-cased model (https://huggingface.co/KB/bert-base-swedish-cased) and have been fine-tuned to solve a multi-label sentiment analysis task.

Acknowledgments

This work has been supported by the National Language Bank of Sweden and the HUMINFRA infrastructure project, both funded by the Swedish Research Council (2017-00626 & 2021-00176) and the project Rumour Mining (MXM19-1161:1) financed by the Bank of Sweden Tercentenary Foundation (Riksbankens Jubileumsfond).

References

Hassan Abdirahman Farah and Arzu Gorgulu Kakisim. 2023. Enhancing Lexicon Based Sentiment Analysis Using n-gram Approach. *Smart Applications with Advanced Machine Learning and Human-Centred Problem Design*. ICAIAME. Engineering Cyber-Physical Systems and Critical Infrastructures, vol 1. Springer, Cham. https://doi.org/10.1007/978-3-031-09753-9_17

Valentin Barriere and Alexandra Balahur. 2020. Improving Sentiment Analysis over non-English Tweets using Multilingual Transformers and Automatic Translation for Data-Augmentation. *Proceedings of the 28th International Conference on Computational Linguistics*. Pages 266–271 Barcelona, Spain (Online).

Steven Bird, Ewan Klein and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Anton Borg and Martin Boldt. 2020. Using VADER sentiment and SVM for predicting customer response sentiment. *Expert Syst. Appl.,* 162. 113746, https://doi.org/10.1016/j.eswa.2020.113746

Nhan Cach Dang, María N. Moreno-García, and Fernando De la Prieta. 2020. Sentiment Analysis Based on Deep Learning: A Comparative Study. Electronics 9:3: 483. 10.3390/electronics9030483

Hutchatai Chanlekha, Wanchat Damdoung and Mukda Suktarachan. 2018. The Development of Semi-automatic Sentiment Lexicon Construction Tool for Thai Sentiment Analysis. *Advances in Intelligent Systems and Computing.* Vol 684. Springer, Cham. https://doi.org/10.1007/978-3-319-70016-8_9

Marcus Gustafsson. 2020. Sentiment analysis for tweets in swedish: Using a sentiment lexicon with syntactic rules. Bachelor's thesis. [Online]. http://www.diva-portal.org/smash/record.jsf?pid=diva2:1391359

Clayton J. Hutto and Eric Gilbert. 2014. A Parsimonious rule-based model for sentiment analysis of social media text. *18th International Conference on Weblogs and Social Media* (ICWSM-14). Stanford, USA.

Jacqueline Kazmaier and Jan H. van Vuuren. 2022. The power of ensemble learning in sentiment analysis. *Expert Systems with Applications*. Vol 187. https://doi.org/10.1016/j.eswa.2021.115819.

Evgeny Kim and Roman Klinger. 2022. A Survey on Sentiment and Emotion Analysis for Computational Literary Studies. *arXiv preprint* https://doi.org/10.48550/arXiv.1808.03137 (v4).

Ching Leong, Lawrence Jin, Dayoung Kim, Jeongbin Kim, Yik Ying Teo and Teck-Hua Ho. 2022. Assessing the impact of novelty and conformity on hesitancy towards COVID-19 vaccines using mRNA technology. *Com Med.* 2:61. https://doi.org/10.1038/s43856-022-00123-6.

Alexander Ligthart, Cagatay Catal and Bedir Tekinerdogan. 2021. Systematic reviews in sentiment analysis: a tertiary study. *Artif Intell Rev* 54, 4997–5053. https://doi.org/10.1007/s10462-021-09973-3

Maria Chiara Martinis, Chiara Zucco, Mario Cannataro. 2022. An Italian lexicon-based sentiment analysis approach for medical applications. *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. https://doi.org/10.1145/3535508.3545594

Antonio Moreno-Ortiz, Chantal Pérez-Hernández and Maria Del-Olmo. 2013. Managing Multiword Expressions in a Lexicon-Based Sentiment Analysis System for Spanish *Proceedings of the 9th Workshop on Multiword Expressions*. Pp. 1-10. Atlanta, USA. https://aclanthology.org/W13-1001

László Nemes and Attila Kiss. 2021. Prediction of stock values changes using sentiment analysis of stock news headlines. *J. Inf Telec*, 5. pp. 375-394, https://doi.org/10.1080/24751839.2021.1874252

Nusrat Jahan Prottasha, Abdullah As Sami, Md Kowsher, Saydul Akbar Murad, Anupam Kumar Bairagi, Mehedi Masud and Mohammed Baz. 2022. Transfer Learning for Sentiment Analysis Using BERT Based Supervised Fine-Tuning. *Sensors* 22:11: 4157. https://doi.org/10.3390/s22114157

Aamir Rashid and Ching-Yu Huang. 2021. Sentiment Analysis on Consumer Reviews of Amazon Products. *J of Comp Theory & Eng*. Vol. 13:2. https://doi.org/10.7763/IJCTE.2021.V13.1287.

Sara Rosenthal, Noura Farra and Preslav Nakov. 2017. SemEval-2017 Task 4: Sentiment Analysis in Twitter. *Proceedings of the 11th International Workshop on Semantic Evaluations* (SemEval-2), pp. 502–518. Vancouver, Canada. c ACL. https://aclanthology.org/S17-2088.pdf

Jacobo Rouces, Lars Borin and Nina Tahmasebi. 2020. Creating an Annotated Corpus for Aspect-Based Sentiment Analysis in Swedish. *Proceedings of the 5th conference in Digital Humanities in the Nordic Countries*. Riga, Latvia. http://ceur-ws.org/Vol-2612/short18.pdf

Irene Russo, Tommaso Caselli, and Carlo Strapparava. 2015. SemEval-2015 task 9: CLIPEval implicit polarity of events. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 443–450, Denver, Colorado. Association for Computational Linguistics.

Karsten Michael Tymann, Matthias Lutz, Patrick Palsbröker and Carsten Gips. 2019. GerVADER - A German adaptation of the VADER sentiment analysis tool for social media texts. *Proceedings of the Conference Lernen, Wissen, Daten, Analysen* (LWDA). Pp. 178-189. Berlin, Germany

Ravi Verma, Amit Chhabra & Ankit Gupta. 2022. A statistical analysis of tweets on covid-19 vaccine hesitancy utilizing opinion mining: an Indian perspective. *Soc. Netw. Anal. Min.* 13, 12. https://doi.org/10.1007/s13278-022-01015-2.

Colex2Lang: Language Embeddings from Semantic Typology

Yiyi Chen
Dep. of Computer Science
Aalborg University
Copenhagen, Denmark
yiyic@cs.aau.dk

Russa Biswas
FIZ Karlsruhe, AIFB
Karlsruhe Inst. of Technology
Karlsruhe, Germany
biswasrussa@gmail.com

Johannes Bjerva
Dep. of Computer Science
Aalborg University
Copenhagen, Denmark
jbjerva@cs.aau.dk

Abstract

In semantic typology, colexification refers to words with multiple meanings, either related (polysemy) or unrelated (homophony). Studies of cross-linguistic colexification have yielded insights into, e.g., psychology, historical linguistics and cognitive science (Xu et al., 2020; Brochhagen and Boleda, 2022; Schapper and Koptjevskaja-Tamm, 2022; Karjus et al., 2021; François, 2022). While NLP research up until now has mainly focused on integrating syntactic typology (Naseem et al., 2012; Täckström et al., 2013; Zhang and Barzilay, 2015; Daiber et al., 2016; de Lhoneux et al., 2018; Ponti et al., 2019; Chaudhary et al., 2019; Üstün et al., 2020; Oncevay et al., 2020; Yu et al., 2021; Ansell et al., 2021; Zhao et al., 2021; Oncevay et al., 2022), we here investigate the potential of incorporating semantic typology, of which colexification is an example. We propose a framework for constructing a large-scale synset graph and learning language representations with node embedding algorithms. We demonstrate that cross-lingual colexification patterns provide a distinct signal for modelling language similarity and predicting typological features. Our representations achieve a 9.97% performance gain in predicting lexico-semantic typological features and expectantly contain a weaker syntactic signal. This study is the first attempt to learn language representations and model language similarities using semantic typology at a large scale, setting a new direction for multilingual NLP, especially for low-resource languages.[1]

1 Introduction

Semantic typology studies cross-lingual semantic categorization (Evans et al., 2010). The term "colexification", which encompasses both polysemy and homophony, was introduced to the field of semantic typology by François (2008). This study focuses on cross-lingual colexification patterns, where the same lexical form is used in distinct languages to express multiple concepts. For instance, *bla* in Monpa Changprong and *afu* in Rikou both express the concepts DUST and ASH (Rzymski et al., 2020).

Colexification was first used in linguistic typology to create semantic maps. Haspelmath (2003) created a semantic map with 12 languages, and François (2008) pointed out that the number of different senses increases with the number and variety of languages used. In recent years, big data, and improved data creation and curation techniques have led to the development of datasets like Concepticon (Forkel et al., 2020), and BabelNet (Navigli and Ponzetto, 2012), which make large-scale cross-lingual semantic comparisons possible. The Cross-Linguistic Colexifications (CLICS) database was created based on the Concepticon collection and is being continuously maintained. The current version[2] CLICS[3] includes 4,228 colexification patterns across 3,156 languages. In this paper, we create a synset graph based on multilingual WordNet (Miller, 1995) data from BabelNet 5.0, compare it with the concept graph extracted from CLICS[3], and explore the impact of data scope on language representation learning.

We hypothesize that language representations learned using semantic typology encapsulate a distinct language signal, and the data size of colexifications has an impact on the learned language representations and the modelled language similari-

[1] GitHub: https://shorturl.at/bioUZ.

[2] https://clics.clld.org/

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 673–684

ties. Importantly, we expect that this type of signal can be used to improve semantically oriented downstream tasks in NLP. To test this hypothesis, we propose a framework `Colex2Lang` (cf. Section 3) to learn language representations leveraging semantic typology, conduct typological feature prediction, and model language similarities. Our experiments on typological feature prediction focus on the domain of *semantic* features, so as to investigate the extent to which a semantic signal is encapsulated by our language representations.

Specifically, we make the following contributions: (i) We generate and evaluate 24 sets of language embeddings based on large-scale colexification databases, using four advanced node embeddings algorithms, i.e., Node2Vec (Grover and Leskovec, 2016), ProNE (Zhang et al., 2019), GGVec [3], and GloVe [4] (Pennington et al., 2014); (ii) we conduct thorough experiments on typological feature prediction to compare colexification-informed and more general language embeddings (Malaviya et al., 2017; Östling and Tiedemann, 2017), which provides a strong benchmark for further research; (iii) we demonstrate the usability of modelling language similarities based on colexification patterns, and argue for the potential of utilising semantic typology in NLP applications.

2 Related Work

Colexification Cross-linguistic colexifications were first formalized by François (2008) for the creation of semantic maps. Semantic maps represent the relation between recurring meaning expressions in a language graphically (Haspelmath, 2003). The basic idea underpinning this method is that language-specific patterns of colexifications indicate semantic closeness or relatedness between the meanings that are colexified (Hartmann et al., 2014). When investigated cross-lingually, colexification patterns can provide insights in various fields, such as recognizing cognitive principles (Berlin and Kay, 1991; Schapper et al., 2016; Jackson et al., 2019; Gibson et al., 2019; Xu et al., 2020; Brochhagen and Boleda, 2022), diachronic semantic shifts in individual languages (Witkowski and Brown, 1985; Urban, 2011; Karjus et al., 2021; François, 2022), and the evolution of language contact (Heine and Kuteva, 2003; Koptjevskaja-Tamm and Liljegren, 2017; Schapper and Koptjevskaja-Tamm, 2022).

Jackson et al. (2019) investigated cross-lingual colexifications in the domain of emotions and found that languages have different associations between emotional concepts. For example, Persian speakers associate the concept of GRIEF with REGRET closely whereas Dargwa speakers associate it with ANXIETY. The cultural variation and universal structure shown in the emotion semantics provide interesting insights into NLP. Di Natale et al. (2021) used colexification patterns to test whether the words linked by colexification patterns capture similar affective meanings, and subsequently expanded affective norms lexica to cover exhaustive word lists when additional data are available. Inspired by Jackson et al. (2019), Sun et al. (2021) proposed emotion semantic distance, measuring how similarly emotions are lexicalized across languages, to improve cross-lingual transfer learning performance on sentiment analysis. Bao et al. (2021) show that there exists no universal colexification pattern by analyzing colexifications from BabelNet, Open Multilingual WordNet (Bond and Foster, 2013), and CLICS[3].

Closely related to our work, Harvill et al. (2022) constructed a synset graph from BabelNet to improve performance on the task of lexical semantic similarity. Instead of modelling only word similarity using colexification patterns, we strive to model language similarity in this study and show that the language embeddings learned on colexification patterns capture a unique semantic signal compared to language embeddings encapsulating syntactical signals. Moreover, we experiment with different node embedding algorithms and compare three colexification datasets. The framework provides a strong benchmark for further investigating how semantic typological aspects of language embeddings can be leveraged in broader applications, especially for low-resource multilingual NLP.

Node Embeddings Node embeddings can be broadly classified into three different categories namely **(i)** matrix factorization-based models, **(ii)** random walk-based models, and **(iii)** deep neural network-based models, as discussed in (Cui et al., 2018).

In matrix factorization-based models, an adjacency matrix is used to denote the topology of a network. Matrix factorization techniques, such as Singular Value Decomposition (SVD) and Non-

[3] `https://github.com/VHRanger/nodevectors`

[4] `https://shorturl.at/myzKR`

negative Matrix Factorization (NMF), can be applied to address this problem. GraRep (Cao et al., 2015) considers k-hop neighbourhoods utilizing SVD of the adjacency matrix. This model often only captures small-order proximity and has a significant computational complexity for large graphs. The asymmetric transitivity is preserved by the HOPE (Ou et al., 2016) model as it converts the problem to a generalised SVD problem reducing the complexity. ProNE (Zhang et al., 2019) introduces a sparse matrix factorization to achieve initial node representations efficiently.

Random walks are used to maintain local neighbourhoods of nodes and their attributes (Newman, 2005), by increasing the likelihood of a node's neighbourhood given its embedding using the Skip-gram model (Mikolov et al., 2013). The objective behind these models is to optimize via stochastic gradient descent on a single-layer neural network, resulting in decreased computing complexity.

DeepWalk (Perozzi et al., 2014) randomly chooses a node and proceeds to walk to each neighbouring node until it reaches its maximum length (or some random length). LINE (Tang et al., 2015) aims to embed nearby vertices that either have linkages between them (optimizing for first-order proximity) or have a shared 1-hop neighbourhood (optimizing for second-order proximity). Node2vec (Grover and Leskovec, 2016) proposes a second-order random walk approach to sample the neighbourhood nodes with biasing parameters of Breadth First Search (BFS) and Depth First Search (DFS). A meta-strategy for graph embedding under recurrent construction of nodes and edges into condensed graphs with the same global structure is proposed by HARP (Chen et al., 2018). These graphs serve as source initializations for the detailed graphs that are embedded, producing appropriate node and edge embeddings as a consequence. Metapath2vec (Dong et al., 2017) is an extension of DeepWalk that formalizes meta-path-based random walks to build a node's neighbourhood, then uses a heterogeneous skip-gram model.

GGVec algorithm directly minimizes distances between the related nodes and is designed for large networks. It uses negative sampling followed by minimization loss to learn the node embeddings based on the minimal dot product of edge weights. Another node embedding model follows the word embedding model GloVe (Pennington et al., 2014) which is based on word co-occurrences and is beneficial for sparse matrices. The graph is represented by an adjacency matrix and the co-occurrence matrix is calculated using the frequency of node co-occurrences in the graph instead of word co-occurrences.

Typological Feature Prediction Linguistic typologists analyse languages in terms of their structural properties (Croft, 2002). As documenting and categorising such cross-lingual variation across the languages in the world is one of the core activities in typology, one of the outcomes of research in linguistic typology is large typological databases (e.g. the World Atlas of Language Structures (WALS, Dryer and Haspelmath (2013)). While such variation can be found across the spectrum of languages, the earliest work in the field largely focused on morphosyntactic properties (e.g. Greenberg (1957)), concretely looking at minimally meaning-bearing elements (morphemes), combinations thereof, and patterns of their use. For instance, well-documented features include word ordering (e.g. English is SVO, Japanese is SOV) and affixation (German uses case suffixes, Berber uses case prefixes).

Prediction of such features has gained interest in recent years (Malaviya et al., 2017; Bjerva et al., 2019a, 2020; Bjerva and Augenstein, 2021), and it has been shown that embeddings trained solely from tasks such as machine translation (Malaviya et al., 2017) or language modelling (Östling and Tiedemann, 2017) can encapsulate such features. Further analysis has shown that the nature of the underlying data used to generate language embeddings can have a significant impact on what features are encapsulated (Bjerva and Augenstein, 2018a,b; Bjerva et al., 2019b), and even that such representations contain typological generalisations (Östling and Kurfalı, 2023). Previous work is limited in that it almost exclusively relates to *morphosyntactic* typological features. In this work, we aim to present initial evidence that a lexico-semantic signal can be better learned from a lexico-semantic data source.

3 Colex2Lang

To better understand and leverage semantic typological features in NLP, we propose a framework – `Colex2Lang` (Fig.1) – to model language representations based on a synset graph,

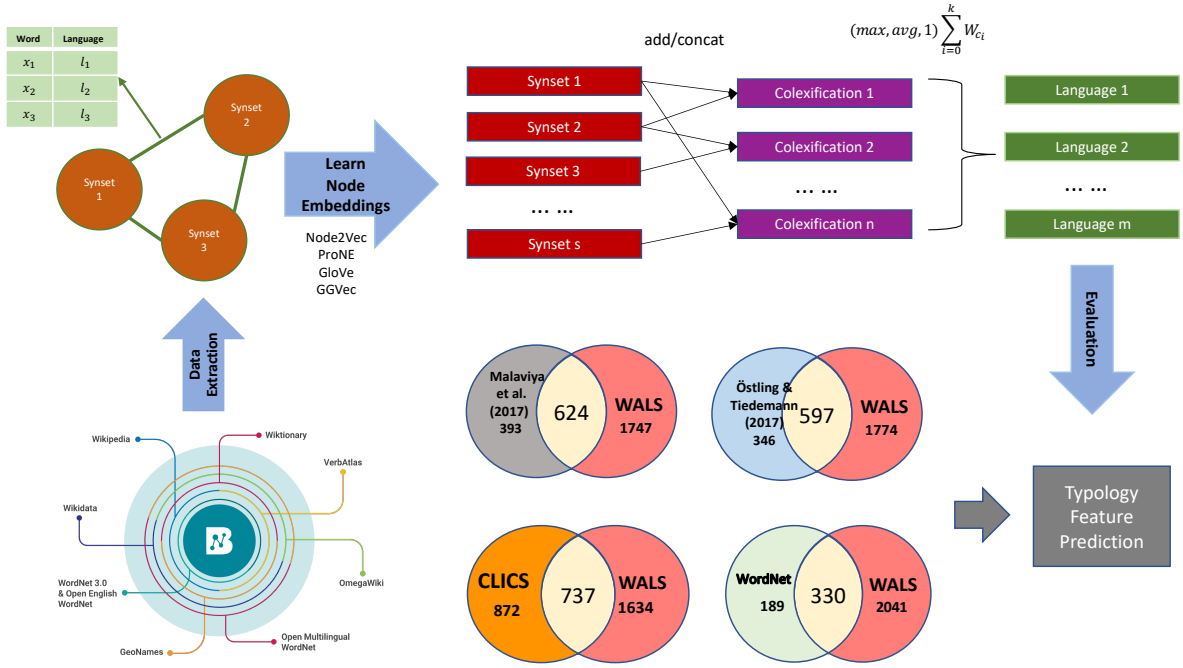


Figure 1: Framework for Colex2Lang. The numbers in the Venn diagrams denote the number of languages.

created from large-scale databases, and evaluate and analyse the language representations. The framework `Colex2Lang` is composed of the following steps:

Building the Synset/Concept Graph

We use WordNet synsets, extracted from BabelNet 5.0, to create a synset graph. The construction of a synset graph is formalized in Harvill et al. (2022) (see details in Appendix A). In BabelNet, every synset is either a concept or a named entity or has no type. The dataset with only concepts and with all types of synsets from WordNet are created, denoted as "WordNet Concept" and "WordNet" respectively. In analogy, CLICS[3] provides a graph of concepts, from which we extracted the colexification patterns for all the languages having an ISO 639-2 code [5], denoted as CLICS. In this study, we use "concept" and "synset" interchangeably. The statistics of the curated datasets are shown in Table 1.

Creating Synset and Language Embeddings To capture the semantic associations among synsets, given the synset/concept graph G_s, we train synset embeddings using four node embedding algorithms and compare them: Node2Vec, ProNE, GGVec and GloVe. Given the learned synset embeddings, we obtain the colexification embeddings W_c by concatenating or summing the synset embeddings W_s; thereafter, the language embeddings W_l are created by summing, averaging or max-pooling the consisting colexification embeddings W_c. For example, if the synset embeddings are trained with ProNE, and are concatenated to compose colexification embeddings, which in turn are max-pooled to obtain language embeddings, we denote the language embeddings as $W_{prone\_concat+max}$.

Evaluation To obtain insights into the learned language embeddings based on the colexification patterns, such as which aspect of language these language embeddings capture and to what extent they can assist in improving NLP tasks, we conduct typological feature prediction and analyse the results in depth. Furthermore, the language embeddings are used to model language similarities, to demonstrate the potential of applications in contributing to cross-lingual transfer learning.

4 Experiments

Datasets To better understand the impact of data scope on the NLP task performance, we curate three different datasets, i.e., WordNet, WordNet Concept, and CLICS, as described in Sec-

[5] `https://shorturl.at/hBCF0`

Dataset	#(C, X, L)	Colexifications (C)	Lexicalizations (X)	Synsets / Concepts	#Language (L) (Pair)
WordNet	6,199,897	2,525,591	974,346	105,827	519 (134421)
WordNet Concept	6,075,413	2,486,485	920,031	99,817	519 (134421)
CLICS	68,560	4,228	53,259	1,647	1609 (332783)

Table 1: Statistics of Colexification Datasets

∩WALS	#Language	Lexicon			Complex Sentences			Nominal Categories			Simple Clauses			10 Feature Areas		
		#F	#V	#D	#F	#V	#D	#F	#L	#D	#F	#V	#D	#F	#V	#D
CLICS	737	13	4	93	7	4	86	29	5	145	26	4	142	188	9	288
WordNet (Concept)	330	13	2	58	7	4	56	29	5	92	26	4	89	185	8	166
Malaviya et al. (2017)	624	13	4	92	7	4	63	29	5	112	26	4	117	190	9	238
Östling and Tiedemann (2017)	597	13	4	85	7	4	60	29	5	103	26	4	109	190	9	219

Table 2: Statistics of Typology Feature Prediction Datasets. Under each feature area and in all ten feature areas, #F represents the total number of features, #V represents the average number of feature values, #D represents the average number of data samples.

tion 3. As shown in Table 1, there are far more (unique) colexification patterns in fewer languages in WordNet-based datasets compared to CLICS, i.e., 6 Mio colexifications with more than 2 Mio unique colexification patterns constructed from 105K synsets in 330 languages, and 68K colexifications with 4K unique colexification patterns from 1,647 concepts across 1609 languages, respectively. The synset embeddings are trained separately on the three datasets with four different node embeddings algorithms, and the language embeddings are composed accordingly, as described in Section 3. Eventually, for each dataset, there are 24 sets of colexification-informed language embeddings [6].

We hypothesize that (i) the colexification-informed language embeddings capture a unique language aspect and (ii) the language embeddings learned on large-scale WordNet datasets present stronger semantic typological signals than the ones trained on CLICS. To test this, we rely on WALS v2020.3 [7], the most used and comprehensive database for typology feature prediction, consisting of 2,662 languages. For our experiment, we extract language data from WALS by ISO 639-2 codes, resulting in a dataset of a total of 2,371 languages, 192 typological features across ten feature areas, i.e., phonology, morphology, lexicon, complex sentences, nominal categories, nominal syntax, simple clauses, verbal categories, word order, and other. To test hypothesis (i), the language embeddings from Malaviya et al. (2017) and Östling and Tiedemann (2017) are used, which are tested for superior performance in typological feature prediction in syntax, phonology and genealogical features, respectively. Specifically, to test hypothesis (ii), we analyse the CLICS and WordNet-based language embeddings' performance on the typology feature prediction and their ability to represent the language similarity compared to typological features (cf. Section 5).

Subsequently, four datasets are created for typology feature prediction by the common set of languages, i.e., CLICS ∩ WALS, WordNet (Concept) ∩ WALS, Malaviya et al. (2017) ∩ WALS, and Östling and Tiedemann (2017) ∩ WALS. The statistics of the intersecting languages with WALS and selecting typological feature areas are shown in Table 2.

Experimental Setup We conduct the typology feature prediction experiments using a simple classifier consisting of a one-layer feedforward neural network with a dropout of 50%, and a softmax layer. For each feature, a multi-class classifier is trained maximally for 100 epochs. The cross-entropy loss is used to evaluate at the end of each epoch. To ensure a fair comparison, for all three datasets, as shown in Table 2, a common set of test data across the data sets is created, consisting of 74 languages. Then for each dataset, the rest of the data is split into train and dev sets. The number of data samples is very limited for each feature, as indicated in Table 2. Ten-fold cross-validation on the train-dev splits is therefore implemented to promote the performance.

To assess whether learned language embeddings capture extra semantic information, we im-

[6] The learned language embeddings are made publicly accessible in our GitHub repository `https://shorturl.at/zFISY`.

[7] `https://doi.org/10.5281/zenodo.7385533`

plement a baseline classifier with a majority vote, and a model with the same one-layer feedforward neural network structure but with an embedding layer initialized with random distribution.

5 Analyses and Results

Comparing Language Embeddings As indicated in Table 2, each feature area has an uneven distribution of features, feature labels and data samples. Hence, the macro F1 score is used to record test results for each feature, and for each feature area, the results of all the included features are averaged. For colexification-informed language embeddings, we present the results of the models with the median and best averaged macro F1-scores for each selecting feature area and the averaged results of all the feature areas.

As shown in Table 3, the baseline and the model with randomly initialized embeddings perform on par across the datasets, whereas all the colexification-informed language embeddings beat the baseline for each feature area and also on average across feature areas, and present the most performance gain in the lexicon area, i.e., 9.91 and 9.97 with WordNet Concept (best) ($W_{glove\_concat+avg}$) and CLICS (best) ($W_{prone\_concat+max}$), respectively. In contrast, the language embeddings from Malaviya et al. (2017) perform the worst for the lexicon features, while having the most performance gain in syntactic feature areas. While the language embeddings from Östling and Tiedemann (2017) perform better in lexicon feature areas compared to Malaviya et al. (2017), both best performing colexification-informed language embeddings still have two percent more performance gains. These results not only corroborate our hypothesis that the colexification-informed language embeddings capture a unique aspect, especially in semantic typological features, but also indicate that in general, leveraging semantic typology information could boost the performance of downstream tasks.

Capturing Lexicon Typological Features To better understand how the colexification-informed language embeddings better capture semantic typological information, we analyze the performance of lexicon feature prediction with several representative examples, as visualized in Figure 2.

The left side of Figure 2 presents the performance of CLICS (best) model and the corresponding Random model in predicting each feature (e.g., Number of Basic Colour Categories), the colour of the circle represents the feature values (e.g., 6-6.5 and 11), and the size of the circles indicates its proportion of the data samples for the regarding values in the train data (e.g., there are more data samples for the feature value "11" than "6-6.5"). Overall, CLICS outperforms Random in almost each feature value across lexicon features. In comparison, CLICS excels at the uneven distribution of train data samples. For instance, for features "Number of Non-Derived Basic Colour Categories" and "Number of Basic Colour Categories", the feature values "4.5" and "11" have fewer samples compared to their counterparts, while Random cannot detect them, CLICS obtained 50% and 80% performance.

Similar results are shown for the Wordnet-based models and their corresponding Random model, as shown on the right side of Figure 2. For the feature "Number of Basic Colour Categories", both WordNet and WordNet Concept models achieve the perfect score compared to the Random counterpart, which is not able to identify the minority class at all. Whereas, WordNet Concept outperforms WordNet for the feature "Number of Non-Derived Basic Colour Categories", WordNet Concept has a 100% macro F1-score with WordNet and Random failing to identify the minority class.

These results demonstrate that the models trained with colexification-informed language embeddings have learned to better capture the semantic typology information compared to randomly initialized embeddings. The language embeddings could be further fine-tuned and applied to assist other NLP applications.

Language Similarities Having attested that the colexification-informed language embeddings capture the semantic typological aspects of languages, we investigate how well the language similarities represented by the semantic typology features and the language embeddings correlate.

To represent languages by lexicon features, we generate a vector for each language by encoding a 13-dimensional vector with the feature values padded with -1, if the feature value is absent. The cosine similarities among the vectors are calculated. Similarly, the cosine similarities are calculated for the language embeddings. The Pearson correlation coefficient and p-value [8] for testing non-correlation are calculated between the lan-

[8] https://shorturl.at/rDO89

Model	Lexicon	Complex Sentences	Nominal Categories	Simple Clauses	Average (All Features)
CLICS ∩ WALS					
Baseline	39.85	21.89	21.94	26.73	29.82
Random	37.88 (-1.97)	23.16 (+1.27)	21.06 (-0.88)	27.14 (+0.41)	29.97 (+0.15)
CLICS (Median)	41.88 (+2.03)	27.73 (+5.84)	26.11 (+4.17)	**29.71 (+2.98)**	30.45 (+0.63)
CLICS (Best)	**49.76 (+9.91)**	**29.32 (+7.43)**	**27.33 (+5.39)**	27.91 (+1.18)	**34.96 (+5.14)**
WordNet ∩ WALS					
Baseline	37.87	19.26	24.12	37.67	33.06
Random	38.54 (+0.67)	19.26	22.42 (-1.70)	34.99 (-2.68)	32.95 (-0.11)
WordNet (Median)	36.59 (-1.28)	23.89 (+4.63)	28.05 (+3.93)	37.12 (-0.56)	34.17 (+1.11)
WordNet Concept (Median)	39.09 (+1.23)	25.43 (+6.17)	27.73 (+3.61)	37.63 (-0.04)	34.94 (+1.88)
WordNet (Best)	47.07 (+9.20)	26.17 (+6.91)	32.56 (+8.44)	**40.23 (+2.56)**	37.11 (+4.05)
WordNet Concept (Best)	**47.84 (+9.97)**	**26.52 (+7.26)**	**34.53 (+10.11)**	38.96 (+1.29)	**39.91 (+6.85)**
Malaviya et al. (2017) ∩ WALS					
Baseline	34.83	18.94	21.98	32.76	31.00
Random	34.83	19.68 (+0.74)	21.21 (-0.77)	33.69 (+0.93)	30.94 (-0.06)
MTCELL	34.43 (-0.4)	0.2549 (+6.55)	**34.74 (+12.76)**	42.79 (10.03)	35.14 (+4.15)
MTVEC	21.85 (-12.98)	23.55 (+4.61)	25.03 (+3.05)	**36.21 (+3.45)**	34.49 (+3.49)
MTBOTH	31.29 (-3.54)	**29.83 (+10.89)**	31.66 (+9.68)	39.37 (+6.61)	**38.21 (+7.22)**
Östling and Tiedemann (2017) ∩ WALS					
Baseline	35.01	18.99	20.39	34.44	31.07
Random	35.01	18.99	21.62(+1.23)	34.57 (+0.13)	30.88 (-0.19)
L1	35.17 (+0.16)	**26.94 (+7.95)**	25.32 (+4.93)	**37.78 (+3.34)**	31.26 (+0.20)
L2	**42.64 (+7.63)**	17.14 (-1.85)	**26.78 (+6.39)**	36.02 (+1.58)	31.80 (+0.73)
L3	34.68 (-0.33)	22.90 (+3.91)	23.99 (+3.60)	35.59 (+1.15)	**33.51 (+2.45)**

Table 3: Test Results of Typological Feature Prediction. Results are in macro-f1 scores, numbers in brackets are the performance gains compared to the corresponding baseline, bold numbers indicate the highest performance gain compared to the corresponding baseline model, and the underlined results indicate the model with the highest performance gain per feature.

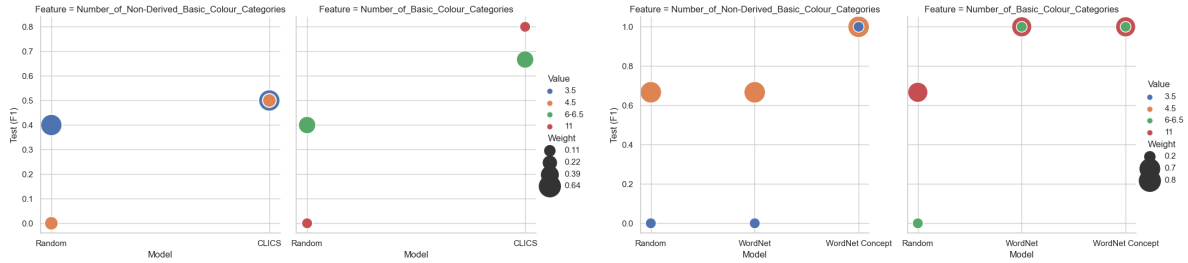


Figure 2: Performance of Predicting Lexicon Typological Features. The test results are in macro F1-scores, the colour of the circle represents the feature values, and the size of the circles indicates the size of the data samples for the regarding values in the train data.

Language Embeddings	#Language (Pair)	Correlation Coefficient (P-value)	#Language (Pair)	Correlation Coefficient (P-value)
CLICS	343 (58653)	- 0.049 (4.436e-33*)	8 (28)	- 0.0876 (0.6575)
WordNet	216 (23220)	**0.1469** (3.525e-112*)	8 (28)	0.7679 (1.838e-06*)
WordNet Concept	216 (23220)	0.1274 (1.339e-84*)	8 (28)	**0.8515** (9.210e-09*)

Table 4: Correlation between Language Similarities represented by Lexicon Typological Features and Colexification-informed Language Embeddings. * indicates that the correlation is statically significant, the numbers in bold indicate the highest correlation coefficients.

guage similarities represented by lexicon typology features and language embeddings. We present the results for the three best-performing language embeddings with both whole language sets intersected with WALS and a case study on a set of Nordic and Baltic languages, as shown in Table 4.

When tested with large sets of language pairs, i.e., 58,653 and 23,220 in CLICS and WordNet-based, respectively, all three correlations are statistically significant, and WordNet-based language embeddings present stronger positive correlations with lexicon typological features in representing language similarities. This verifies our hypothesis that the language embeddings learned on large-scale WordNet datasets present stronger semantic typological signals than the one trained on CLICS.

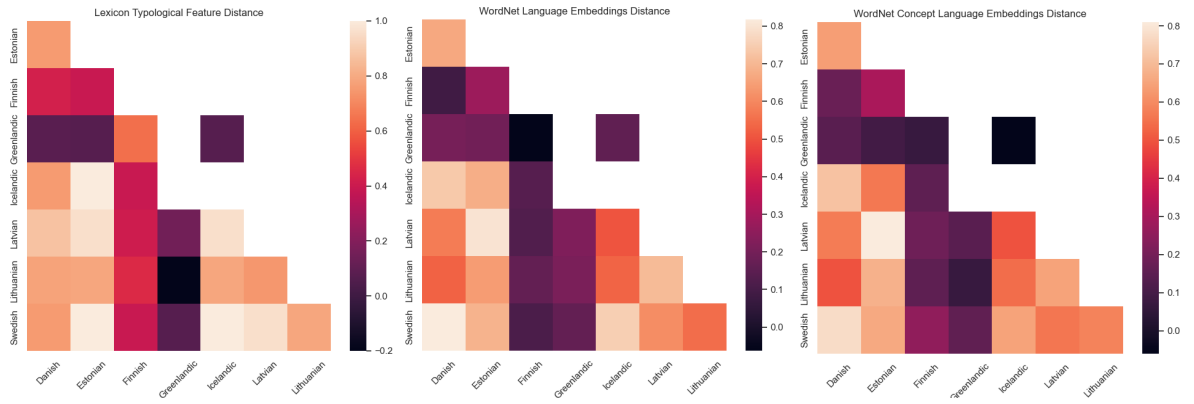


Figure 3: Language similarities represented by Lexicon Typological Features and Colexification-informed Embeddings.

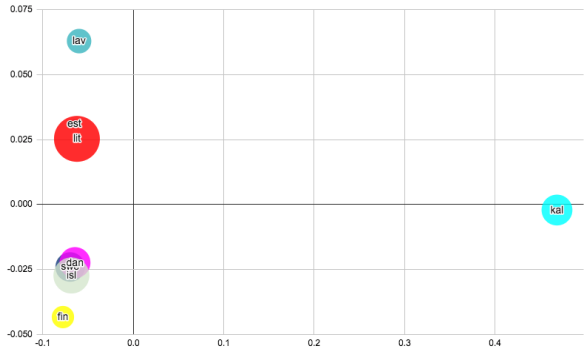


Figure 4: Language similarities represented by applying PCA on WordNet language embeddings.

The information density of the language embeddings increases with the number of incorporated synsets and colexification patterns.

A set of Nordic and Baltic languages are selected to compare further the represented language similarities. Both WordNet-based language embeddings present strong positive correlations, i.e., 0.7679 and 0.8515, respectively, and the correlations are statistically significant, as shown on the right side of Table 4. To further analyse the results, the heatmap is used to visualize the language similarities represented by lexicon features and WordNet-based language embeddings, as shown in Figure 3. The most distinctive difference is that Finnish is highly similar in terms of lexicon features compared to other languages but relatively dissimilar in terms of WordNet-based language embeddings. In this context, the WordNet-based embeddings arguably present a more realistic image of language similarities semantically.

We differentiated the WordNet Concept from WordNet dataset, assuming that a dataset with only concepts would avoid data noises and render language embeddings able to better capture the semantic associations between languages. However, the analysed results do not corroborate the assumption. On the contrary, the language embeddings learned on all the WordNet synsets present a stronger correlation (+0.02) with lexicon typological features.

To further investigate language similarities, we apply PCA to the WordNet-based language embeddings (Figure 4). We can observe that, e.g., Scandinavian languages are clustered together, as expected. Another observation is that Finnish is relatively close to this cluster, owing to a relatively high amount of overlapping colexification patterns from language contact with Swedish, as compared to Estonian which is placed closer to one of its contact languages, Lithuanian.

6 Conclusion and Future Work

In this study, we have proposed a framework `Colex2Lang` to leverage colexifications to learn language representations and explored the potential of using semantic typology in NLP. A large-scale synset graph is constructed using WordNet source from Babelnet, and three datasets of colexification are processed including CLICS. Subsequently, within each dataset, 24 language embeddings variants are learned, and further evaluated and analysed by typology feature prediction and modelling language similarity. We have demonstrated, at a large scale, that colexification-informed language embeddings capture a distinctive aspect of languages in terms of semantic typology, and the data scope of the curated synsets

affects the performance of applying language embeddings. Furthermore, the analysis of representing language similarities by using learned language embeddings illustrates a realistic approach.

A large body of research has demonstrated the use of syntactic, genealogical and geographical information from linguistic typology to learn language representations, model language similarities, and further improve transfer learning performance in downstream tasks in NLP. Our work is the first attempt to learn language representations and model language similarity by leveraging semantic typology. The framework provides a strong benchmark for further research in this direction.

For future work, the benefits of applying colexification-informed language embeddings will be extensively explored. Multilingual semantic parsing is a clear candidate, where a cross-lingual signal based on colexifications may prove useful. The language similarities represented by colexifications could further inspire multilingual transfer learning, as in leveraging high-resource languages with dedicated lexical data to improve performance in semantically similar low-resource languages.

Acknowledgments

This work is supported by the Carlsberg Foundation under a *Semper Ardens: Accelerate* career grant held by JB, entitled 'Multilingual Modelling for Resource-Poor Languages', grant code *CF21-0454*. We are furthermore grateful to Heather Lent for her insightful comments on earlier versions of this manuscript, and to Esther Ploeger for assistance in extracting typological feature prediction data from WALS.

References

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. MAD-G: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hongchang Bao, Bradley Hauer, and Grzegorz Kondrak. 2021. On universal colexifications. In *Proceedings of the 11th Global Wordnet Conference*, pages 1–7, University of South Africa (UNISA). Global Wordnet Association.

Brent Berlin and Paul Kay. 1991. *Basic color terms: Their universality and evolution*. Univ of California Press.

Johannes Bjerva and Isabelle Augenstein. 2018a. From phonology to syntax: Unsupervised linguistic typology at different levels with language embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 907–916.

Johannes Bjerva and Isabelle Augenstein. 2018b. Tracking typological traits of uralic languages in distributed language representations. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, pages 76–86.

Johannes Bjerva and Isabelle Augenstein. 2021. Does typological blinding impede cross-lingual sharing? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 480–486, Online. Association for Computational Linguistics.

Johannes Bjerva, Yova Kementchedjhieva, Ryan Cotterell, and Isabelle Augenstein. 2019a. Uncovering probabilistic implications in typological knowledge bases. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3924–3930, Florence, Italy. Association for Computational Linguistics.

Johannes Bjerva, Robert Östling, Maria Han Veiga, Jörg Tiedemann, and Isabelle Augenstein. 2019b. What do language representations really represent? *Computational Linguistics*, 45(2):381–389.

Johannes Bjerva, Elizabeth Salesky, Sabrina J Mielke, Aditi Chaudhary, Celano Giuseppe, Edoardo Maria Ponti, Ekaterina Vylomova, Ryan Cotterell, and Isabelle Augenstein. 2020. Sigtyp 2020 shared task: Prediction of typological features. In *The Second Workshop on Computational Research in Linguistic Typology*, pages 1–11. Association for Computational Linguistics.

Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362.

Thomas Brochhagen and Gemma Boleda. 2022. When do languages use the same word for different meanings? the goldilocks principle in colexification. *Cognition*, 226:105179.

Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 891–900.

Aditi Chaudhary, Elizabeth Salesky, Gayatri Bhat, David R. Mortensen, Jaime Carbonell, and Yulia Tsvetkov. 2019. CMU-01 at the SIGMORPHON 2019 shared task on crosslinguality and context in morphology. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 57–70, Florence, Italy. Association for Computational Linguistics.

Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. 2018. Harp: Hierarchical representation learning for networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

William Croft. 2002. *Typology and Universals*. Cambridge University Press.

Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. 2018. A survey on network embedding. *IEEE transactions on knowledge and data engineering*, 31(5):833–852.

Joachim Daiber, Miloš Stanojević, and Khalil Sima'an. 2016. Universal reordering via linguistic typology. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3167–3176, Osaka, Japan. The COLING 2016 Organizing Committee.

Anna Di Natale, Max Pellert, and David Garcia. 2021. Colexification networks encode affective meaning. *Affective Science*, 2(2):99–111.

Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Nicholas Evans et al. 2010. Semantic typology. In *The Oxford handbook of linguistic typology*. Oxford University Press.

Robert Forkel, Sebastian Bank, Christoph Rzymski, and Hans-Jörg Bibiko. 2020. clld/clld: clld - a toolkit for cross-linguistic databases.

Alexandre François. 2008. Semantic maps and the typology of colexification. *From polysemy to semantic change: Towards a typology of lexical semantic associations*, (106):163.

Alexandre François. 2022. Lexical tectonics: Mapping structural change in patterns of lexification. *Zeitschrift für Sprachwissenschaft*, 41(1):89–123.

Edward Gibson, Richard Futrell, Steven P. Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy. 2019. How efficiency shapes human language. *Trends in Cognitive Sciences*, 23(5):389–407.

Joseph H Greenberg. 1957. The nature and uses of linguistic typologies. *International Journal of American Linguistics*, 23(2):68–77.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Iren Hartmann, Martin Haspelmath, and Michael Cysouw. 2014. Identifying semantic role clusters and alignment types via microrole coexpression tendencies. *Studies in Language. International Journal sponsored by the Foundation "Foundations of Language"*, 38(3):463–484.

John Harvill, Roxana Girju, and Mark Hasegawa-Johnson. 2022. Syn2Vec: Synset colexification graphs for lexical semantic similarity. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5259–5270, Seattle, United States. Association for Computational Linguistics.

Martin Haspelmath. 2003. The geometry of grammatical meaning: Semantic maps and cross-linguistic comparison. In *The new psychology of language*, pages 217–248. Psychology Press.

Bernd Heine and Tania Kuteva. 2003. On contact-induced grammaticalization. *Studies in Language. International Journal sponsored by the Foundation "Foundations of Language"*, 27(3):529–572.

Joshua Conrad Jackson, Joseph Watts, Teague R. Henry, Johann-Mattis List, Robert Forkel, Peter J. Mucha, Simon J. Greenhill, Russell D. Gray, and Kristen A. Lindquist. 2019. Emotion semantics show both cultural variation and universal structure. *Science*, 366(6472):1517–1522.

Andres Karjus, Richard A Blythe, Simon Kirby, Tianyu Wang, and Kenny Smith. 2021. Conceptual similarity and communicative need shape colexification: An experimental study. *Cognitive Science*, 45(9):e13035.

Maria Koptjevskaja-Tamm and Henrik Liljegren. 2017. *Semantic Patterns from an Areal Perspective*, Cambridge Handbooks in Language and Linguistics, page 204–236. Cambridge University Press.

Miryam de Lhoneux, Johannes Bjerva, Isabelle Augenstein, and Anders Søgaard. 2018. Parameter sharing between dependency parsers for related languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4992–4997, Brussels, Belgium. Association for Computational Linguistics.

Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language*

Processing, pages 2529–2535, Copenhagen, Denmark. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 629–637, Jeju Island, Korea. Association for Computational Linguistics.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Mark EJ Newman. 2005. A measure of betweenness centrality based on random walks. *Social networks*, 27(1):39–54.

Arturo Oncevay, Duygu Ataman, Niels Van Berkel, Barry Haddow, Alexandra Birch, and Johannes Bjerva. 2022. Quantifying synthesis and fusion and their impact on machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1308–1321.

Arturo Oncevay, Barry Haddow, and Alexandra Birch. 2020. Bridging linguistic typology and multilingual machine translation with multi-view language representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2391–2406, Online. Association for Computational Linguistics.

Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Edoardo Maria Ponti, Ivan Vulić, Ryan Cotterell, Roi Reichart, and Anna Korhonen. 2019. Towards zero-shot language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2900–2910, Hong Kong, China. Association for Computational Linguistics.

Christoph Rzymski, Tiago Tresoldi, Simon J Greenhill, Mei-Shin Wu, Nathanael E Schweikhard, Maria Koptjevskaja-Tamm, Volker Gast, Timotheus A Bodt, Abbie Hantgan, Gereon A Kaiping, et al. 2020. The database of cross-linguistic colexifications, reproducible analysis of cross-linguistic polysemies. *Scientific data*, 7(1):1–12.

Antoinette Schapper and Maria Koptjevskaja-Tamm. 2022. Introduction to special issue on areal typology of lexico-semantics. *Linguistic Typology*, 26(2):199–209.

Antoinette Schapper, Lila San Roque, and Rachel Hendery. 2016. *12. Tree, firewood and fire in the languages of Sahul*, pages 355–422. De Gruyter Mouton, Berlin, Boston.

Jimin Sun, Hwijeen Ahn, Chan Young Park, Yulia Tsvetkov, and David R. Mortensen. 2021. Cross-cultural similarity features for cross-lingual transfer learning of pragmatically motivated tasks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2403–2414, Online. Association for Computational Linguistics.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1061–1071, Atlanta, Georgia. Association for Computational Linguistics.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077.

Matthias Urban. 2011. Asymmetries in overt marking and directionality in semantic change. *Journal of Historical Linguistics*, 1(1):3–47.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online. Association for Computational Linguistics.

Stanley R. Witkowski and Cecil H. Brown. 1985. Climate, clothing, and body-part nomenclature. *Ethnology*, 24(3):197–214.

Yang Xu, Khang Duong, Barbara C. Malt, Serena Jiang, and Mahesh Srinivasan. 2020. Conceptual relations predict colexification across languages. *Cognition*, 201:104280.

Dian Yu, Taiqi He, and Kenji Sagae. 2021. Language embeddings for typology and cross-lingual transfer learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7210–7225, Online. Association for Computational Linguistics.

Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. 2019. Prone: Fast and scalable network representation learning. In *IJCAI*, volume 19, pages 4278–4284.

Yuan Zhang and Regina Barzilay. 2015. Hierarchical low-rank tensors for multilingual transfer parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1857–1867, Lisbon, Portugal. Association for Computational Linguistics.

Wei Zhao, Steffen Eger, Johannes Bjerva, and Isabelle Augenstein. 2021. Inducing language-agnostic multilingual representations. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 229–240, Online. Association for Computational Linguistics.

Robert Östling and Murathan Kurfalı. 2023. Language embeddings sometimes contain typological generalizations.

Robert Östling and Jörg Tiedemann. 2017. Continuous multilinguality with language vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 644–649.

A Construction of Colexification Graph

We adopt the algorithm presented in Harvill et al. (2022) to construct a large-scale synset graph from WordNet synsets for our study (cf. Section 3). The difference in our approach lies in the addition of G_s at line 3 and line 9, as shown in Algorithm 1. G_s affords the constructions of colexification embeddings and language embeddings after obtaining synset embeddings trained on G with node embeddings algorithms (cf. Section 3).

Algorithm 1 Construction of Colexification Graph: Given a set of languages L and corresponding vocabularies V, create graph edges between all colexified synset pairs (nodes), consisting of the set of tuples of lemmas and their language.

1: **function** CONSTRUCTGRAPH(L,V)
2: $CSP \leftarrow \{\}$ ▷ Colexified Synset Pairs
3: $G_s \leftarrow \mathbf{graph}$
4: **for** $l \in L$ **do**
5: **for** $x \in V_l$ **do**
6: **if** $|S_x| \geq 2$ **then**
7: **for** $\{s_1, s_2\} \in \binom{S_x}{2}$ **do**
8: $CSP \leftarrow CSP \cup \{s_i, s_j\}$
9: $G_s(s_1, s_2) \leftarrow \{x, l\}$
10: **end for**
11: **end if**
12: **end for**
13: **end for**
14: $G \leftarrow \mathbf{graph}$
15: **for** $s_1, s_2 \in CSP$ **do**
16: $G(s_1, s_2) \leftarrow 1$
17: **end for**
18: **return** G
19: **return** G_s
20: **end function**

Toxicity Detection in Finnish Using Machine Translation

Anni Eskelinen, Laura Silvala, Filip Ginter, Sampo Pyysalo and Veronika Laippala
TurkuNLP
University of Turku, Finland
aeeske, laura.s.silvola, figint, sampo.pyysalo, mavela
@utu.fi

Abstract

Due to the popularity of social media platforms and the sheer amount of user-generated content online, the automatic detection of toxic language has become crucial in the creation of a friendly and safe digital space. Previous work has been mostly focusing on English leaving many lower-resource languages behind. In this paper, we present novel resources for toxicity detection in Finnish by introducing two new datasets, a machine translated toxicity dataset for Finnish based on the widely used English Jigsaw dataset and a smaller test set of Suomi24 discussion forum comments originally written in Finnish and manually annotated following the definitions of the labels that were used to annotate the Jigsaw dataset. We show that machine translating the training data to Finnish provides better toxicity detection results than using the original English training data and zero-shot cross-lingual transfer with XLM-R, even with our newly annotated dataset from Suomi24.

1 Introduction

Social media is filled with moderated and unmoderated content with foul language such as threats, insults and swears. Due to the popularity of the platforms and the sheer amount of comments, posts and other user-generated content they include, moderation by human-raters is getting impossible. This makes automatic toxicity detection a requirement in the monitoring of social media platforms and other online settings in order to guarantee a safe and friendly digital space.

In recent years, many studies have tackled the detection of toxic language as well as other similar and relevant tasks, such as the detection of hate speech and offensive language (Davidson et al., 2017; MacAvaney et al., 2019). However, most datasets and thus most of the studies focus on English, leaving other languages with very scarce resources (Davidson et al., 2017; Androcec, 2020). At the same time, the development of the resources, in particular the creation of manually annotated training data, is very time-consuming. Cross-lingual transfer learning has offered a solution to this challenge by allowing the use of data in one language to predict examples in another one. This method has showed promising results in tasks such as register labeling (Rönnqvist et al., 2021; Repo et al., 2021) and offensive language detection (Pelicon et al., 2021). Additionally, recent advances in machine translation open up the question of how to use machine translation to do the language transfer and create novel resources for a language.

In this paper, we address the lack of resources for toxicity detection in languages other than English by benefitting from the recent advances in machine translation. Specifically, we present the first publicly available dataset for toxicity detection in Finnish that we develop by machine translating the English Jigsaw Toxicity Dataset that is claimed to be the biggest and most widely used toxicity dataset (Androcec, 2020). We show that machine translating the dataset to Finnish provides better results for toxicity detection than cross-lingual transfer learning, where a cross-lingual XLM-R model (Conneau et al., 2020) is fine-tuned using the original English Jigsaw training set and tested on the Finnish machine translated test set. Furthermore, to test how much machine translation modifies the content of the dataset and thus causes performance loss, we backtranslate the dataset from Finnish to English, demonstrating only a minimal decrease in performance. Finally, in order to examine how much toxic content the trained model identifies from another source

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 685–697

than the Wikipedia edit comments included in Jigsaw, we create another test set for toxicity detection in Finnish by manually annotating comments from the Finnish discussion forum Suomi24 and building a dataset of 2,260 comments. The annotations follow the label description guidelines that were used to annotate the original English dataset. We show that while the model does identify toxic content also from the discussion forum comments, the change of text source does present some challenges.

As machine translation systems, we test two systems to see whether there are major differences to our results: the DeepL machine translation service[1] and Opus-MT (Tiedemann and Thottingal, 2020), see Section 3.2. The DeepL machine translated dataset, the native Finnish dataset and the resulting fine-tuned FinBERT large model are all openly available at the TurkuNLP Huggingface page[2].

2 Related Work

Toxicity, in terms of speech, text or behaviour, is an umbrella term that encompasses many kinds of language use, such as hate speech, abusive language, offensive language and harmful language. In this paper, we follow the definition adopted in the Jigsaw dataset and define toxicity as "rude, disrespectful, or unreasonable language that is likely to make someone leave a discussion" (Jigsaw).

Toxicity detection (van Aken et al., 2018; Pavlopoulos et al., 2020; Burtenshaw and Kestemont, 2021) is related to many other similar classification tasks such as hate speech detection (Davidson et al., 2017; MacAvaney et al., 2019) and offensive language classification (Davidson et al., 2017; Jahan and Oussalah, 2020). In all these tasks, the goal is to identify harmful text in, e.g., social media, where comments can be flagged for review or automatically deleted.

Toxicity datasets and datasets for other related tasks are mostly monolingual with English being the most popular—most studies have used the same Jigsaw dataset that we use (Androcec, 2020). For instance, Carta. et al. (2019) reported ROC_AUC-scores of nearly 90% on this dataset. Additionally, datasets are available, e.g., for Spanish (Androcec, 2020), and a multilingual dataset has been developed as a part of the Kaggle competition on multilingual toxicity detection[3].

The available datasets represent various domains and text lengths, ranging from short Twitter posts (Davidson et al., 2017) to Wikipedia editor comments featured by the Jigsaw dataset we are using, see Section 3.1. Similarly, the annotation strategies vary from multi-label annotation where one instance can have several independently assigned labels to multi-class where one instance can be assigned just one label (Davidson et al., 2017) and to even a binary setting where each instance is either clean or toxic (D'Sa et al., 2020). Due to these differences, combining several datasets to increase the number of examples in training data is difficult.

Similarly, the subjectivity entailed in toxicity creates a challenge for its automatic detection—as people interpret things differently, a single correct interpretation of a message as toxic or not may not exist (see discussion in Ross et al. (2016)). In addition to model performance, the subjectivity can be noted in low inter-annotator agreements. For instance, Waseem (2016) reported a kappa of .57, which can be interpreted as *weak*.

Cross-lingual zero-shot transfer learning where the model is trained on one language and tested on another relies on multilingual language models that have been trained on massive amounts of multilingual data (Conneau et al., 2020; Devlin et al., 2018). These have been used for the zero-shot cross-lingual transfer of hate speech detection and offensive/abusive language detection. For instance, Pelicon et al. (2021) report that a multilingual BERT-based classifier achieves results that are comparable to monolingual classifiers in offensive language detection and also Eronen et al. (2022) demonstrate that zero-shot cross-lingual transfer can achieve competitive results for abusive language detection. However, Nozza (2021) note also challenges—the zero-shot transfer of hate speech detection can be complicated by non-hateful, language-specific taboo interjections that are interpreted by the model as signals of hate speech, and Leite et al. (2020) also found that zero-shot transfer did not produce accurate results for toxicity detection in Brazilian Portuguese.

Machine translation can be considered a mode of transfer learning that has become viable with the advances of natural language processing. In

[1] https://www.deepl.com/translator

[2] https://huggingface.co/TurkuNLP

[3] https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification

particular, the method has been used in toxic language detection to get more data by data augmentation (Rastogi et al., 2020) and by translating data to English to be able to use ready-made models (Kobellarz and Silva, 2022). Kobellarz and Silva (2022) found that comments that were analyzed as toxic in Portuguese were not as toxic when translated to English—however, the same behaviour may not apply to other language pairs. To our knowledge, no experiments comparing cross-lingual transfer by a multilingual model and by machine translation have been made previously.

3 Data and Translation

3.1 Jigsaw Toxicity Dataset

The data used in this paper is the Jigsaw dataset developed by Google and released as a Kaggle competition[4]. The dataset is based on comments from Wikipedia's talk page edits and consists of 223,549 comments. The dataset collection was done by crowd-sourcing. No specific information about the annotation process is given.

The annotation scheme is composed of six classes: *toxicity*, *severe toxicity*, *identity attack, insult, obscene* and *threat*. Toxicity is a general label encompassing all toxicity and is defined as "rude, disrespectful, or unreasonable language that is likely to make someone leave a discussion", and severe toxicity as "a very hateful, aggressive, disrespectful comment or otherwise very likely to make a user leave a discussion or give up on sharing their perspective". For other definitions, please see the annotation guidelines for Perspective API (Perspective, a,b).

The annotation is set up as multi-label, where each comment annotated as toxic has one or more labels assigned to it. The label distribution of the dataset is presented in Table 1. In total, only 11% of the comments are annotated with at least one of the toxic labels, the rest being left without labels and considered as neutral or non-toxic. This means that the label distribution is highly unbalanced, which, however, comes from the nature of the data as most comments are neutral in discussions. More information about label co-occurrence is given in Figure 1, showing that in particular *obscene* and *insult* as well as *toxicity*, *insult* and *obscene* co-occur.

[4]https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/

	Train	Test
Toxicity	15,924	6,090
Severe toxicity	1,595	367
Threat	478	211
Obscene	8,449	3,691
Insult	7,877	3,427
Identity attack	1,405	712
No label	143,346	57,735

Table 1: Label distribution in the Jigsaw Toxicity Dataset. As each comment may have up to six labels, the total number of labels exceeds the number of comments in the dataset.

The dataset is split into train and test sets with stratified sampling (159,571 and 63,978 comments) following the original Kaggle release. Furthermore, for our training purposes with the Finnish data, a development set is split from the train set by doing stratified splitting and taking 20% of the train set comments.

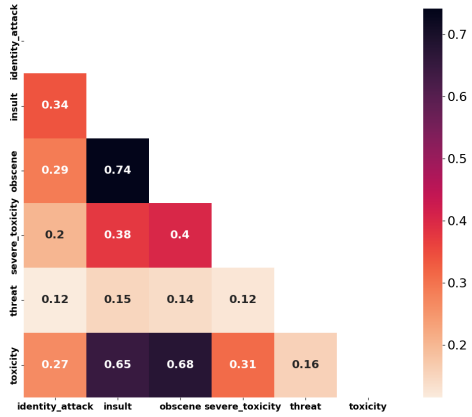


Figure 1: Correlation matrix of the labels of the original train dataset calculated with Pearson standard correlation coefficient. Small values close to zero indicate no correlation between the labels, while higher values closer to 1 suggest correlation and that the labels tend to appear together.

3.2 Jigsaw Toxicity Data in Finnish

We machine translated the original English Jigsaw dataset to Finnish using two translation tools: the DeepL machine translation service[5] and Opus-MT (Tiedemann and Thottingal, 2020). For DeepL, the dataset was converted to the required `.docx`-format for the translation and then back to `.jsonl` after the translation. The English-

[5]https://www.deepl.com/translator

Finnish translation cost less than 100 dollars. None of the comments were lost during this process—a possibility that needs to be considered when transforming data from a format to another.

For Opus-MT, the texts needed to be sentence split, because the tool can only translate one sentence at a time. This was done using the Udpipe REST api[6]. The model used for translation was `Helsinki-NLP/opus-mt-tc-big-en-fi`[7]. Some of the comments from the test set did not survive the translation, as they were not in English. These were edited to only include the notion "EMPTY".

Finally, to examine the loss of performance caused by the machine translation, we also **back-translated** the dataset translated with DeepL from Finnish back to English. This was done using the same method as the English-Finnish translations.

The DeepL machine translated dataset is available at the TurkuNLP Huggingface[8].

3.3 Native Finnish Toxicity Dataset

To examine how much toxic content the fine-tuned model can identify in comments featuring another text variety than the Wikipedia editor comments included in Jigsaw, we developed a new manually annotated test set sampled from Suomi24—the largest online discussion forum in Finland compiled into a giga-size corpus. As the label distribution is very skewed in the Jigsaw dataset with a large majority of comments not annotated for toxicity (see Section 3.1), the sampling was done in a specific manner to ensure a representative set of comments featuring varying degrees of toxicity and the six toxicity classes.

Specifically, we first classified 945,867 comments taken from Suomi24 using a model that was at the time our best performing model which was a fine-tuned base model of FinBERT (Virtanen et al., 2019). Then, for each of the six toxicity labels, we binned the comments to ten bins based on the classifier score for that label (0.0-0.1, 0.1-0.2, ... 0.9-1.0). The distribution of comments in these bins is presented in Appendix A, showing that the classifier is very certain about most of its decisions. In particular, the 0.0-0.1 bins are extremely large, while another set of peaks can be seen on the right end with high scores.

After the binning, we selected randomly 50 comments from each bin for annotation. This gave 500 comments of broadly varying degrees of predicted toxicity for each of the six toxicity labels. Each of the six batches of 500 comments were annotated for one toxicity label only. Thus, the annotations are multi-class instead of the original Jigsaw multi-label, although 23 individual comments were selected in two different batches due to the sampling for each label being independent. This also means that a comment can have some other type of toxicity that was not annotated for that specific comment.

	Label	No label
Toxicity	158	193
Severe toxicity	25	328
Threat	40	391
Obscene	170	239
Insult	145	219
Identity attack	131	221
Total	669	1591

Table 2: Label distribution in the native Finnish annotations.

The annotation was done independently by three native Finnish speakers with borderline cases jointly resolved and documented. This process resulted in guidelines which include general directions for the labels meaning the guidelines can be used for any language as a starting point for annotation. For the initial process of annotating a label, we annotated 100-200 comments and used the definitions of the labels found in the Perspective API (Perspective, a,b) as a starting point, after which we had a discussion where we added our own specifications to the guidelines. Then the last 300-400 comments were annotated according to those guidelines.

The inter-annotator agreement for the initial annotation and the annotations done after the discussion can be found in Table 3. As can be seen, the unanimous agreement is very low in almost every label category, which is common for toxicity datasets as mentioned in Section 2. *Threat* is the only label with a higher agreement of around 80% whereas most of the other labels range between 47 and 66%. Unfortunately, our mean agreement did not get better after the discussion which once again shows the difficulty of the task.

The final dataset was formed using only the comments that were initially unanimously labeled

[6]https://lindat.mff.cuni.cz/services/udpipe/api-reference.php

[7]https://huggingface.co/Helsinki-NLP/opus-mt-tc-big-en-fi

[8]https://huggingface.co/datasets/TurkuNLP/jigsaw_toxicity_pred_fi

	Initial	After discussion
Toxicity	58%	54%
Severe toxicity	63%	66%
Threat	82%	80.3%
Obscene	69%	62%
Insult	47.5%	49.6%
Identity attack	54.5%	66.6%
Mean	62.3%	63%

Table 3: Unanimous inter-annotator agreement (IAA) for the native Finnish toxicity dataset

or for which the label was resolved in a subsequent discussion. While the initial annotations showed significant divergence, this filtering protocol should assure the internal consistency and validity of the dataset. Altogether, the final dataset consists of 2,260 comments natively written in Finnish, further described in Table 2. The guidelines created during the annotation process are published together with the dataset on Huggingface[9].

4 Fine-tuning

We use both monolingual and multilingual state-of-the-art models in the detection experiments. Specifically, the monolingual models are the large and cased versions of the original BERT for English (Devlin et al., 2018) and FinBERT for Finnish (Virtanen et al., 2019). For the crosslingual experiments, we use XLM-RoBERTA (XLM-R) Large (Conneau et al., 2020) because it has been shown to provide better results than the multilingual BERT for many tasks (Repo et al., 2021; Rönnqvist et al., 2021).

All the experiments are done in a multi-label setting. However, when evaluating classifier performance on the native Finnish test set where the comments are only annotated for one toxicity label at a time, we ignore other labels than the one annotated in the batch. Furthermore, we made a custom loss function to the model, giving the labels weights in order to tackle the imbalanced label distribution in the data. The weights were calculated based on the labels' frequency in the training data. The resulting weights make the labels with fewer examples in the training data more important to the model and labels with the most examples receive a lower importance. E.g., *threat* received a weight of 47.6901 due to it appearing in the data only 478 times and *toxicity* the weight 1.4905 due to appearing 15924 times in the data.

No pre-processing for the texts was done to get the best results since previous studies had found that with deep learning pre-processing can make the results worse (Saeed et al., 2018).

For training, we used sequence length of 512 by truncating at the end and did hyperparameter optimization with grid search using learning rate (LR) of (1e-5..5e-5), batch size of (4, 8, 12), and epochs (10) with early stopping and evaluation every 2500 steps. All the hyperparameters were optimized on the development set. For the crosslingual experiments with XLM-R, we optimized on the English development set and tested on the translated Finnish test set. The best hyperparameters can be found in Appendix B. Furthermore, we used threshold optimization to find the best threshold that maximizes the results for the F1-score.

As metrics in the evaluation, we use micro precision and recall, micro-F1, macro-F1 and ROC_AUC. Precision shows how many of the positive predictions are correct, and recall how many of all the positive cases in the data were found. F1-score is the balanced and harmonic mean of precision and recall. Micro-F1 specifically calculates metrics globally and macro-F1 for each label separately, finding their unweighted mean. Thus, macro-F1 does not take label imbalance into account.

ROC_AUC score is the Area Under the Receiver Operating Characteristic Curve. This metric was used for the scoring of the Kaggle competition held for the original dataset, although only done on the probabilities and 90% of the data as opposed to us using the thresholded label and the full test set.

The codebase for fine-tuning can be found on Github[10] and the fine-tuned model can also be found on Huggingface[11].

5 Results

5.1 Translation and Transfer

The results of the toxicity detection experiments using the original English and the translated datasets are presented in Table 4. As a baseline, we can consider the results of the English BERT model, 0.69 F1-score (micro-avg.) and 0.89

[9]https://huggingface.co/datasets/TurkuNLP/Suomi24-toxicity-annotated

[10]https://github.com/TurkuNLP/toxicity-classifier

[11]https://huggingface.co/TurkuNLP/bert-large-finnish-cased-toxicity

Model	Train	Test	Precision	Recall	F1-micro	FI-macro	ROC_AUC
BERT	En	En	0.59	0.81	0.69	0.61	0.89
FinBERT	Fi-DeepL	Fi-DeepL	0.58	0.76	0.66	0.57	0.87
FinBERT	Fi-Opus-MT	Fi-Opus-MT	0.57	0.77	0.65	0.57	0.88
XLM-R	Fi-DeepL	Fi-DeepL	0.56	0.76	0.65	0.57	0.87
XLM-R	En	Fi-DeepL	0.60	0.54	0.57	0.47	0.76
XLM-R	Fi-DeepL+En	Fi-DeepL	0.56	0.78	0.65	0.57	0.88
BERT	Backtr-En	En	0.59	0.77	0.67	0.60	0.87

Table 4: Results with different language pairs and models.

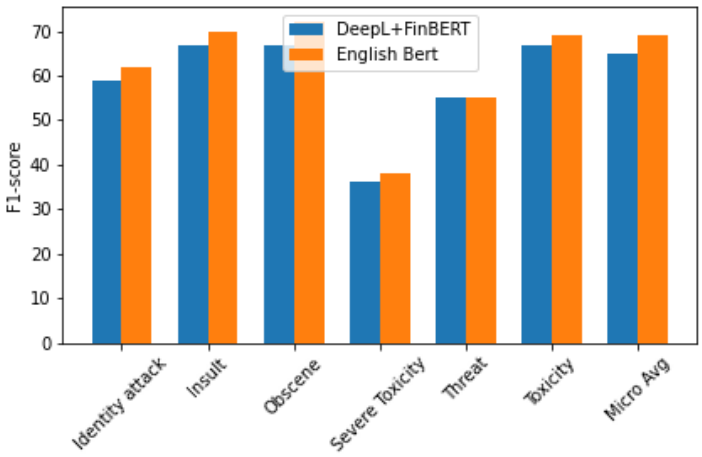


Figure 2: Class-specific F1-scores.

ROC_AUC, trained and tested on the original English data. This is very similar to the results reported by Carta. et al. (2019) using the same Jigsaw dataset and BERT (see Section 2).

FinBERT trained and tested on the machine translated data performs numerically slightly worse than BERT on the English data: 0.66 F1-score with DeepL and 0.65 with OPUS-MT. The loss of performance is, however, very small. With this result, we decide to run the further experiments with the data translated with DeepL.

The multilingual XLM-R performs numerically very similarly to FinBERT with the Finnish DeepL-translated data: 0.65 F1-score. However, its performance is clearly lower when trained on English and only tested on Finnish: 0.57 F1-score. Thus, our results support those by Leite et al. (2020), who noted that zero-shot transfer from English to another language can be challenging.

Our results thus suggest that circumventing the language barrier provides much better results with machine translation than with a cross-lingual model. The quality of the machine translations is further supported by the results on the backtranslated English dataset. By showing only a 2% loss in the F1-score, this experiment supports the quality of the translations.

Even combining the original English data and its DeepL-translations in the training set does not provide better results than training and testing on the DeepL-translated Finnish data alone, and the model trained and tested in English outperforms both of these settings. This can suggest that transfer, done either with a model or machine translation, can have some effect on the results.

Given the subjectivity associated with toxicity detection, and the IAA scores discussed in Section 2 and our own IAA scores in Section 3.3, the detection results are very close to what can be expected for this task. Additionally, for practical purposes, it is noteworthy that the recall is approximately 20% higher than the precision for all the experiments except for the cross-lingual one. When used for cleaning data or moderating a platform, false positives can be less dangerous than false negatives. This further consolidates the practical usability of the method.

5.2 Label-Specific Scores

Nozza (2021) showed that language-specific differences in, e.g., taboo expressions can challenge cross-lingual toxicity detection. These differences

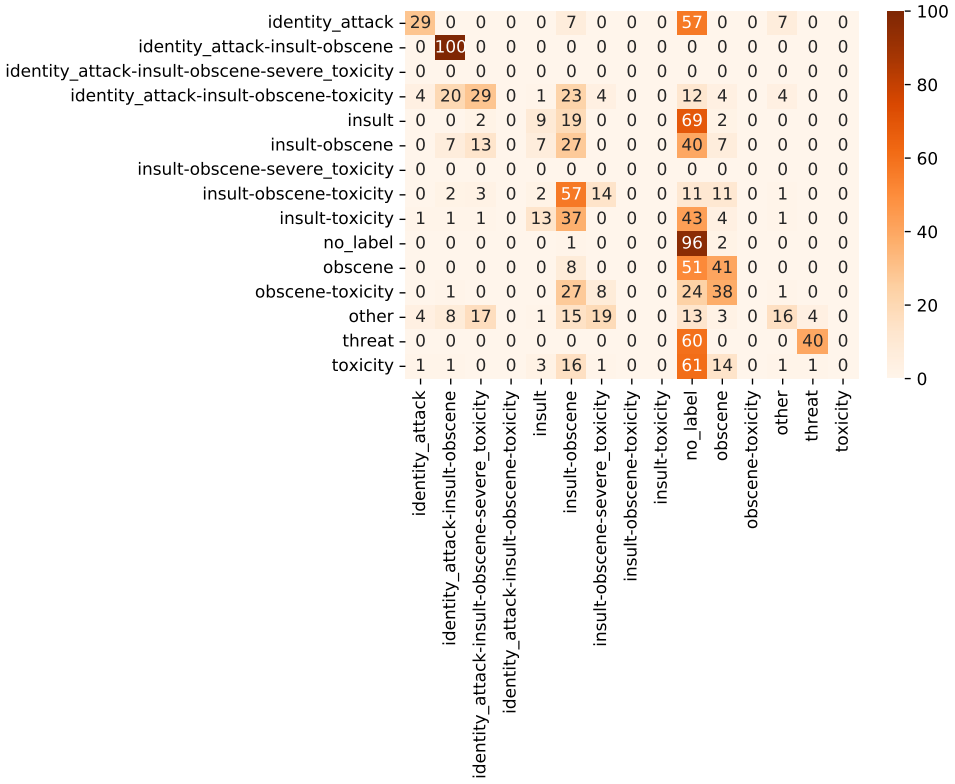


Figure 3: Most frequent classes (rows) and their misclassifications (columns), as percentages of the total number of instances in the data. For the sake of simplicity, co-occurring labels have been fixed as multi-class.

may lead to lower results in particular for some subtypes of toxicity. To ensure that the cross-lingual results we presented in Section 5.1 are not affected by these or similar issues, we inspect label-specific performance metrics. We focus on two models: the best-performing Finnish model trained using FinBERT and the DeepL-translated data, as well as the English model trained using BERT and the original English data.

Figure 2 presents the label-specific metrics obtained using the two models. First, we can see that while the global scores obtained by the English model are slightly higher, the difference remains similar across the labels. Thus, the language transfer does not seem to affect them unevenly.

However, the F1-scores do vary between the different labels. In particular, *severe toxicity* and *threat* receive lower detection scores than the other labels. *Threat* is a very infrequent label, which may also explain its lower detection rate. However, the size of *severe toxicity* is similar to *identity attack*, which nevertheless received better performance. This suggests that the label as such is more vague and less well-defined linguistically.

5.3 Error Analysis of the Translated Dataset

Figure 3 presents to which labels the most frequent labels and label combinations were predicted to using the best-performing FinBERT model trained on the DeepL-translated data, and Table 5 presents examples of misclassified comments[12].

The most frequent misclassification for many labels and their combinations is *no label*. In part, this is because of the frequency of comments without any label in the training. On the other hand, this can also reflect the gray zone between different kinds of toxic comments and the ones annotated without a label, prone to subjective interpretation. This can be seen in Example 2 in Table 5, annotated as *insult* and predicted without a label. Whether *hyödytön* 'useless' actually is an insult or not can be prone to subjective interpretations. Furthermore, the Perspective API annotation guidelines in fact note that insult should be targeted towards a person (Perspective, a,b).

Similarly, *severe toxicity* as a label is eas-

[12]The examples may be considered offensive. They do not reflect the views of the authors and exclusively serve to explain the linguistic characteristics of the comments.

ID	Text	True label(s)	Predicted label(s)
FI-1	*Senkin pikku paskiainen == Senkin pikku paskiainen*	insult, obscene	insult, obscene, toxicity
EN-1	You little son of a bitch == You little son of a bitch	severe toxicity, toxicity	
FI-2	*Wikipedia on täysin hyödytön*	insult	no label
EN-2	Wikipedia is completely useless		
FI-3	*Painu helvettiin, pedo. Nyljen sinut elävältä, kastelen sinut bensiinillä ja poltan sinut kuoliaaksi, kusipää. Syön vitun sydämesi, senkin pikku mulkku.*	insult, obscene threat, toxicity	insult, obscene severe toxicity, threat toxicity
EN-3	Fuck off paedo. I am going to fucking skin you alive, douse you in petrol then fucking burn you to death, fucktard. I am going to eat your fucking heart you little prick.		

Table 5: Examples of misclassified and correctly predicted labels in the DeepL-translated data.

ily not predicted. For instance, *insult-obscene-severe toxicity-toxicity* is frequently misclassified as *insult-obscene-toxicity*. Example 1 illustrates this as well—*severe toxicity* is present in the annotations but not predicted. In this case, the error may be caused by the translation, as the Finnish translation is not as toxic as the original English comment and can even be used to communicate affection. As we mentioned in Section 5.2, *severe toxicity* also received relatively low class-specific scores. Figure 3 shows that it is frequently misclassified as simple *toxicity*. For instance, the label combinations *identity attack-insult-severe toxicity* and *insult-obscene-severe toxicity* are frequently confused with the same labels co-occurring with *toxicity*. Examples 1 and 3 illustrate this as well, as *severe toxicity* is erroneously not predicted for Example 1 and is predicted for Example 3, where it should not have been predicted and the correct label would have been just plain *toxicity* with the other labels.

	Prec	Rec	F1
FinBERT-DeepL	0.57	0.59	0.58
FinBERT-DeepL Weighted	0.61	0.74	0.67
XLMR-En	0.50	0.40	0.45
XLMR-En Weighted	0.50	0.40	0.45

Table 6: Micro evaluation results for the native Finnish dataset using threshold 0.5.

5.4 Native Finnish Dataset

We tested the two best-performing models (FinBERT trained on Fi-Deepl and XLM-R trained on the original English data) on the native Finnish Suomi24 annotations in order to examine the model performances on texts featuring different language use than the Wiki edit comments included in Jigsaw. The results are presented in Table 6, showing that while the models do find toxic content from the Suomi24 discussions, the performances decrease in comparison with the original Jigsaw data (see Section 5.1). Nevertheless, similar to our findings with the Jigsaw data, cross-lingual transfer using a multilingual model provides lower results than a monolingual model trained on translations. Further, similar to the Jigsaw dataset, *severe toxicity* and *threat* received low class-specific scores due to the low amount of examples for those classes in the training data. The metrics for the labels can be found in Appendix C.

A reason for the lower metrics on the Suomi24 discussions can be found in the way the data were sampled (see Section 3.3). By taking even samples from all the prediction score bins even though the large majority of the comments were included in the bins with 0-0.1 or 0.9-1 scores, our sampling method emphasized borderline cases (see Appendix A), and the vast majority of the comments the classifier was certain about were disregarded. The metrics do not take into account this imbalance by default and thus, they can be interpreted rather as macro-average known to display low results for skewed data. Therefore, we counted also the *weighted* metrics using the counts of the bins as weights for the true positive, true negative, false positive and false negative counts. The results achieved using FinBERT-DeepL and this weighing are very similar to those achieved with FinBERT-DeepL on Jigsaw (see Section 5.1).

Table 7 shows examples from the native Finnish dataset. Example 4 presents a comment annotated as *no label*, derived from the bin 0.8-0.9 for *identity attack* predicted by a previous model as a very certain *identity attack* and then later labeled by the new large model as *toxicity*, most likely because the model simply associates 'gay' with toxicity. This illustrates the oversensitivity of the model and bias. A case can be made for the text being intended as an insult but without context that is impossible to say. Example 5 shows a comment binned in very certain *identity attack*, annotated

ID	Text	Bin	True label(s)	Pred. label(s)
FI-4	*Oletko mahdollisesti homoseksuaali?*	identity attack	no label	toxicity
EN-4	Are you possibly gay?	0.8-0.9		
FI-5	*jos nämä muslimit saavat räjäytellä pommejaan missä haluavat ympäri maailmaa niin miksemme me saa julkaista vitsikkäitä kuvia.*	identity attack	identity attack	identity attack, toxicity
EN-5	if these Muslims can explode their bombs anywhere they want so why can't we publish funny pictures?	0.9-1		
FI-5	*Tästä tulee iso hitti!*	toxicity	no label	no label
EN-5	This is going to be a big hit!	0.0-0.1		

Table 7: Examples of misclassified and correctly predicted labels in the native Finnish data.

with the same label and then predicted as *identity attack-toxicity*. Here the model succeeds in finding the correct label. Finally, Example 6 presents a comment annotated and predicted as *no label* from the 0-0.1 bin for toxicity—the kind of comment of which the classifier is certain about and our annotation agrees.

6 Conclusion

In this paper, we have presented novel resources for Finnish toxicity detection, and we have shown that machine translation is a viable option for circumventing the language barrier for this task. FinBERT and the DeepL-translated data outperformed XLM-R trained on English and tested on Finnish clearly, and the quality of the translation was further confirmed with the backtranslation experiment, showing only a minimal loss in the original English performance. Thus, our results support previous findings by Isbister et al. (2021) and Kobellarz and Silva (2022). Additionally, our results were also confirmed by the results from the native Finnish test set where translation received better results than transfer and our weighted numbers were comparable with the results from using the original translated test set.

The use of machine translation is a cost-effective alternative for building resources when there is no annotated data available in the target language. However, translation can also cause subtle changes in the meaning, which can result in misclassifications and wrong interpretations. Our analysis showed that the toxicity entailed in the original comment can change during the translation to a much less toxic meaning. Therefore, it is crucial that the effect of the translation is evaluated separately for each language and task.

Furthermore, we acknowledge that our model might feature some bias, as illustrated in Section 3.3. Jigsaw has also reported this—the models may learn to incorrectly associate toxicity with, e.g., identities that frequently co-occur with toxic content. This has led to the creation of a new dataset called "Jigsaw Unintended Bias in Toxicity Classification" [13].

In the future, we should further inspect the possible biases the models developed in this study may feature, as well as the model generalizability. Furthermore, multilingual toxicity detection involving code-switching would offer an interesting avenue for the future. Finally, considering the promising results achieved in this study, the use of machine translation for other tasks and language pairs should certainly be analyzed further.

Acknowledgments

We thank Academy of Finland for financial support and wish to acknowledge CSC – IT Center for Science, Finland for computational resources.

References

Betty van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. 2018. Challenges for toxic comment classification: An in-depth error analysis. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 33–42, Brussels, Belgium. Association for Computational Linguistics.

Darko Androcec. 2020. https://doi.org/10.2478/ausi-2020-0012 Machine learning methods for toxic comment classification: a systematic review. *Acta Universitatis Sapientiae, Informatica*, 12:205–216.

Ben Burtenshaw and Mike Kestemont. 2021. A Dutch dataset for cross-lingual multilabel toxicity detection. In *Proceedings of the 14th Workshop on Building and Using Comparable Corpora (BUCC 2021)*, Online (Virtual Mode). INCOMA Ltd.

Salvatore Carta., Andrea Corriga., Riccardo Mulas., Diego Reforgiato Recupero., and Roberto Saia.

[13] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

2019. A supervised multi-class multi-label word embeddings approach for toxic comment classification. In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KDIR,*, pages 105–112. INSTICC, SciTePress.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ashwin Geet D'Sa, Irina Illina, and Dominique Fohr. 2020. Towards non-toxic landscapes: Automatic toxic comment detection using DNN. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, Marseille, France. European Language Resources Association (ELRA).

Juuso Eronen, Michal Ptaszynski, Fumito Masui, Masaki Arata, Gniewosz Leliwa, and Michal Wroczynski. 2022. Transfer language selection for zero-shot cross-lingual abusive language detection. *Information Processing &amp Management*, 59(4):102981.

Tim Isbister, Fredrik Carlsson, and Magnus Sahlgren. 2021. Should we stop training more monolingual models, and simply use machine translation instead? In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Md Saroar Jahan and Mourad Oussalah. 2020. Team oulu at SemEval-2020 task 12: Multilingual identification of offensive language, type and target of Twitter post using translated datasets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1628–1637. International Committee for Computational Linguistics.

Jigsaw. https://jigsaw.google.com/the-current/toxicity/ Jigsaw toxicity [online].

Jordan K. Kobellarz and Thiago H. Silva. 2022. Should we translate? evaluating toxicity in online comments when translating from portuguese to english. New York, NY, USA. Association for Computing Machinery.

João Augusto Leite, Diego Silva, Kalina Bontcheva, and Carolina Scarton. 2020. Toxic language detection in social media for Brazilian Portuguese: New dataset and multilingual analysis. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, Suzhou, China. Association for Computational Linguistics.

Sean MacAvaney, Yao Hao-Ren, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. 2019. Hate speech detection: Challenges and solutions. *PLoS One*, 14(8).

Debora Nozza. 2021. Exposing the limits of zero-shot cross-lingual hate speech detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Online. Association for Computational Linguistics.

John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. 2020. Toxicity detection: Does context really matter? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4296–4305. Association for Computational Linguistics.

Andraž Pelicon, Ravi Shekhar, Matej Martinc, Blaž Škrlj, Matthew Purver, and Senja Pollak. 2021. Zero-shot cross-lingual content filtering: Offensive language and hate speech detection. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, Online. Association for Computational Linguistics.

Perspective. https://developers.perspectiveapi.com/s/about-the-api-attributes-and-languages?language=en_US Perspective api attributes and languages [online].

Perspective. https://developers.perspectiveapi.com/s/about-the-api-training-data?language=en_US Perspective api training data [online].

Chetanya Rastogi, Nikka Mofid, and Fang-I Hsiao. 2020. Can we achieve more with less? Exploring data augmentation for toxic comment classification.

Liina Repo, Valtteri Skantsi, Samuel Rönnqvist, Saara Hellström, Miika Oinonen, Anna Salmela, Douglas Biber, Jesse Egbert, Sampo Pyysalo, and Veronika Laippala. 2021. Beyond the English web: Zero-shot cross-lingual and lightweight monolingual classification of registers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, Online. Association for Computational Linguistics.

Samuel Rönnqvist, Valtteri Skantsi, Miika Oinonen, and Veronika Laippala. 2021. Multilingual and zero-shot is closing in on monolingual web register classification. In *Proceedings of the 23rd Nordic*

Conference on Computational Linguistics (NoDaLiDa), Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2016. Measuring the reliability of hate speech annotations: The case of the european refugee crisis.

Hafiz Hassaan Saeed, Khurram Shahzad, and Faisal Kamiran. 2018. Overlapping toxic sentiment classification using deep neural architectures. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1361–1366.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for finnish. *CoRR*, abs/1912.07076.

Zeerak Waseem. 2016. Are you a racist or am I seeing things? annotator influence on hate speech detection on Twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 138–142, Austin, Texas. Association for Computational Linguistics.

Appendices

A The distribution of predicted scores for the Suomi24 data before sampling

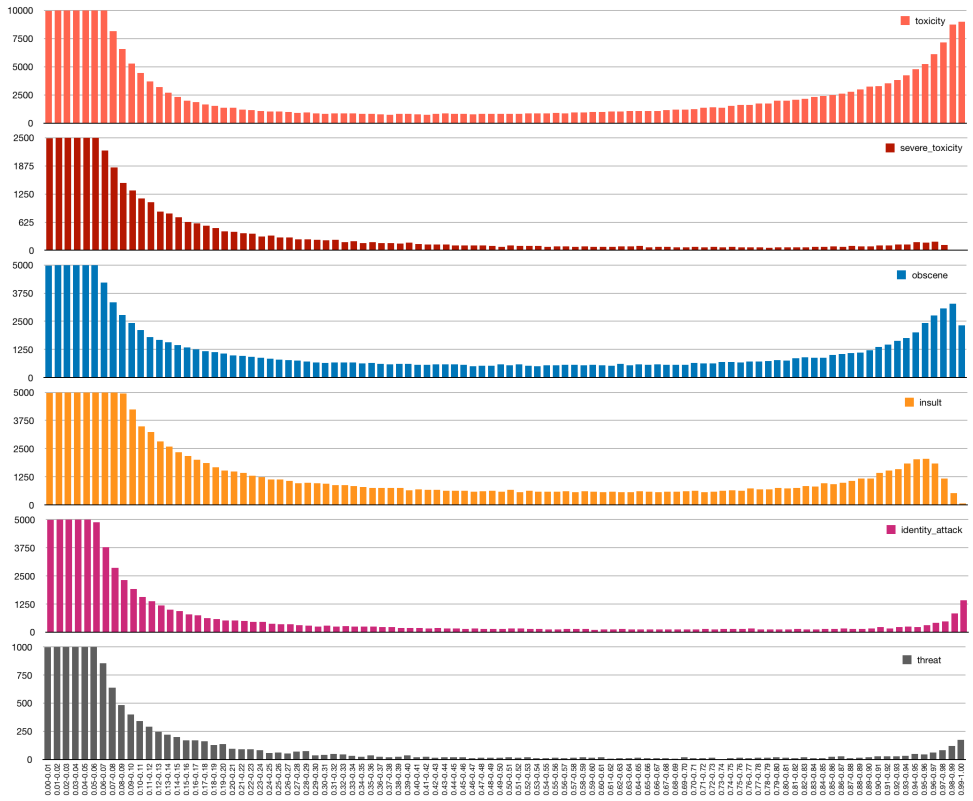


Figure 4: Distribution of prediction scores by label for the Suomi24 data from which our native Finnish dataset examples were sampled for annotation.

B Best hyperparamaters for the trained models

Model	Train	Test	LR	Batch size
BERT	En	En	1e-5	12
FinBERT	Fi-DeepL	Fi-DeepL	2e-5	12
FinBERT	Fi-Opus-MT	Fi-Opus-MT	1e-5	12
XLM-R	Fi-DeepL	Fi-DeepL	1e-5	12
XLM-R	En	Fi-DeepL	1e-5	12
XLM-R	Fi-DeepL+En	Fi-DeepL	1e-5	12
BERT	Backtr-En	En	2e-5	12

Table 8: Best hyperparameters for each model. Constant parameters were epochs 10 and early stopping 5. Threshold for the labels varied due to threshold optimization during training and evaluation.

C Label specific precision, recall and F1 for the native Finnish dataset

Label	Precision	Recall	F1
Identity attack	0.73	0.32	0.45
Insult	0.59	0.47	0.52
Obscene	0.64	0.82	0.72
Severe toxicity	0.12	0.29	0.17
Threat	0.32	0.29	0.30
Toxicity	0.60	0.79	0.69

Table 9: Micro evaluation results for the labels of the native Finnish dataset using FinBERT-DeepL and a threshold of 0.5.

Evaluating a Universal Dependencies Conversion Pipeline for Icelandic

Þórunn Arnardóttir[1], Hinrik Hafsteinsson[1], Atli Jasonarson[2],
Anton Karl Ingason[1], Steinþór Steingrímsson[2]
[1]University of Iceland, [2]The Árni Magnússon Institute for Icelandic Studies
{thar, hinhaf, antoni}@hi.is,
{atli.jasonarson, steinthor.steingrimsson}@arnastofnun.is

Abstract

We describe the evaluation and development of a rule-based treebank conversion tool, UDConverter, which converts treebanks from the constituency-based PPCHE annotation scheme to the dependency-based Universal Dependencies (UD) scheme. The tool has already been used in the production of three UD treebanks, although no formal evaluation of the tool has been carried out as of yet. By manually correcting new output files from the converter and comparing them to the raw output, we measured the labeled attachment score (LAS) and unlabeled attachment score (UAS) of the converted texts. We obtain an LAS of 82.87 and a UAS of 87.91. In comparison to other tools, UDConverter currently provides the best results in automatic UD treebank creation for Icelandic.

1 Introduction

The Universal Dependencies (UD) project is a multilingual project, consisting of dependency treebanks in 138 languages (Zeman et al., 2022; Nivre et al., 2020). UDConverter is a tool which converts a phrase structure treebank to a UD treebank (Arnardóttir et al., 2020), and has been used for creating three UD corpora. Originally configured for Icelandic, the converter can be extended to convert treebanks in languages other than Icelandic, as has been done for a Faroese treebank (Arnardóttir et al., 2020), but it has not been thoroughly evaluated until now. Without such evaluation, the benefit of using the converter is uncertain. Therefore, we manually corrected a portion of a treebank created with the converted UD treebank and evaluate the conversion by comparing the converted sentences' output to the manually corrected ones. The evaluation is used to guide further development of UDConverter, resulting in an improved conversion pipeline.

The paper is structured as follows. Section 2 discusses relevant resources, including UD corpora and methods of creating them. Section 3 describes the evaluation setup used while Section 4 discusses the results, including initial results before the converter was improved. We compare the converter's accuracy scores to the accuracy of three UD parsers in Section 5 and finally, we conclude in Section 6.

2 Background

UDConverter is a Python module for converting bracket-parsed treebanks in the format of the Penn Parsed Corpora of Historical English (PPCHE) to the Universal Dependencies framework (Arnardóttir et al., 2020). It was created in order to convert the Icelandic Parsed Historical Corpus (IcePaHC) (Rögnvaldsson et al., 2012) to the UD CoNLL-U format and has been used for creating three UD corpora, UD_Icelandic-IcePaHC, UD_Icelandic-Modern and UD_Faroese-FarPaHC, all included in version 2.11 of Universal Dependencies (Zeman et al., 2022). The converter takes an original IcePaHC-format tree and converts it to a UD tree, displayed in the CoNLL-U format. As discussed in Arnardóttir et al. (2020), the converter can be extended to convert treebanks in other languages than Icelandic, as long as the input treebanks are in a format similar to the IcePaHC one. The converter's output generally adheres to UD annotation guidelines but no formal evaluation of the converter has been carried out until now.

The UD corpora which were created by using UDConverter were all converted from pre-existing constituency treebanks. These treebanks were manually annotated according to the PPCHE annotation scheme (Kroch and Taylor, 2000; Kroch et al., 2004), which uses labeled bracketing in the same way as the Penn Treebank (Taylor et al., 2003).

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 698–704

This IcePaHC annotation scheme was used as a basis for the rule sets of UDConverter.

UD_Icelandic-Modern was converted from 21st-century additions to IcePaHC, consisting of modern Icelandic texts (Rúnarsson and Sigurðsson, 2020). It contains genres not previously found in the original IcePaHC (Wallenberg et al., 2011), extracted from the Icelandic Gigaword Corpus (Steingrímsson et al., 2018).

Two UD parsers have recently been released for Icelandic (Jasonarson et al., 2022a,b), both of which utilize information from a pre-trained BERT-like model, in this case an ELECTRA model that was pre-trained on Icelandic texts (Daðason and Loftsson, 2022). One of the models was trained with DiaParser (Attardi et al., 2021), an extended version of the Biaffine parser (Dozat and Manning, 2017), which uses contextualized embeddings, as well as attentions, from a transformer model as its input features. The other one was trained with COMBO (Klimaszewski and Wróblewska, 2021), which accepts pre-trained embeddings from a transformer, as well as character and lemma embeddings, in addition to part-of-speech tags, as its input features. Both parsers were trained on two Icelandic UD corpora, UD_Icelandic-IcePaHC and UD_Icelandic-Modern.

3 Evaluation

In order to evaluate UDConverter, we set up a testing experiment where output CoNLL-U files from the converter were manually evaluated and corrected per strict annotation guidelines. These were then compared to the original raw output files. As these files contain identical texts, this enabled a one-to-one comparison, with the manually corrected files serving as a gold standard.

In our evaluation, we focused on measuring the accuracy of the conversion when it comes to heads and dependency relations. For this project, we chose to source sentences for manual correction from the UD_Icelandic-Modern corpus, discussed in Section 2, which then became the test set. In total, 651 sentences of the corpus, 15,140 tokens in total, were manually corrected, out of 80,395 tokens overall. Two annotators with a background in linguistics worked on the manual correction. Sentences were corrected to adhere to annotation rules used in the Icelandic Parallel Universal Dependencies (PUD) corpus (Jónsdóttir and Ingason, 2020), which is the only Icelandic UD corpus which was created manually. The corpus was used as a guideline when UDConverter was developed. The annotators worked on separate sentences, and therefore information on inter-annotator agreement is not available. It would be beneficial to have information on the agreement, but the annotators discussed any uncertainties and came to joint conclusions.

We used a labeled attachment score (LAS) to evaluate the converter, evaluating CoNLL-U output based on how many tokens have been assigned both the correct syntactic head and the correct dependency relation (Kübler et al., 2009). This simple accuracy score corresponds to a labeled F_1 score of syntactic relations. Similar to this score is the unlabeled attachment score (UAS), which evaluates the number of correct heads but does not take the dependency relations into account.

4 Results

Our results show that the converter achieves an LAS of 82.87 and a UAS of 87.91. Our results indicate that the overall error rate of the conversion is not affected by sentence length, with the relationship between sentence length and total errors per sentence being more or less linear. If sentence length is a rough indicator of syntactic complexity, this means that the converter handles complex syntactic structures just as well as simpler ones. This is expected, as the converter works off of a fixed rule set for a given language, which looks at the already annotated phrase structure of the input sentences.

4.1 Initial results

The first evaluation of the converter showed worse results, with an LAS of 72.82 and a UAS of 80.79. After analyzing the difference in the converter's output and the manually corrected texts, a few systematic errors were identified, which accounted for a large proportion of errors. Three of these items related to an incorrect head of a dependent with a particular dependency relation, and two related to an incorrect dependency relation.

Head-related errors

The three head-related errors have the dependency relations *punct*, *cop* and *cc*. *Punct* is used to denote punctuation and was dependent on an incorrect head in 75.63% of cases. An important error relating to *punct* was in the case of end-of-sentence punctuation, which should be dependent on the root of the sentence. 66.28% of *punct* dependency relations dependent on an incorrect head were end-of-

sentence punctuation, i.e. punctuation marks which should have been dependent on the sentence's root, but were for some reason not.

The second head-related error was the *cop* dependency relation, with a 21.86% error rate. This relation is used for copulas, which in Icelandic is the verb *vera* 'be'. Copular constructions are structurally different from other verbal constructions, so this construction had to be handled specifically, marking the predicate as the root of a sentence and the copular verb as its dependent. Determining which word or phrase is the predicate is not always unequivocal, so a copular verb is in some cases dependent on the incorrect word.

The third and final head-related error was the *cc* dependency relation, which was dependent on an incorrect head in 18.52% of cases. This relation is used for a coordinating conjunction and is part of a conjunction phrase in IcePaHC. In a simple example, a conjunction phrase is made up of three words, e.g. two nouns with a coordinating conjunction between them, linking them together. Initially, the converter marks the first noun as the head of the phrase and the conjunction and the second noun as its dependents. According to the UD annotation guidelines, the conjunction should be dependent on the second noun, so this is corrected in the conversion algorithm as part of a series of checks after the initial conversion is done, making the second noun the head of the conjunction. In more complex cases, this correction can go wrong, resulting in the conjunction (*cc*) being dependent on an incorrect head.

Incorrect dependency relations

The two most frequent incorrect dependency relations were *acl* and *obl*. The *acl* relation stands for finite and non-finite clauses that modify a nominal. It had an error rate of 72.01% and was, in most cases, supposed to be replaced by the *xcomp* relation, which denotes an open clausal complement of a verb or an adjective. This error was caused by a fault in the rules of UDConverter, wherein the *acl* relation was incorrectly used for heads of certain subcategories of infinitival clauses, e.g. direct speech, degree infinitives and subjectival infinitives. These clauses are labeled *IP-INF* in the IcePaHC annotation scheme, and this relation was incorrectly mapped to *acl* instead of *xcomp*. These errors were therefore simple to correct.

The second incorrect dependency relation, *obl*, had an error rate of 26.44%. The *obl* relation is used for a nominal which functions as an oblique argument or adjunct. A proportion of these errors are due to the fact that the *obl:arg* relation is used in the manually corrected sentences, but not in the converter. *obl:arg* is a subcategory of the *obl* relation, and is used to distinguish oblique arguments from adjuncts, which have the *obl* relation. This relation was used to have our manually corrected sentences better conform to the Icelandic PUD corpus, which uses this relation.

These five items were analyzed, e.g. how often a relation which should have been *xcomp* was incorrectly *acl*, and a projection was created on the converter's possible LAS if these errors were fixed altogether. This projected LAS is 85.34, which is considerably higher than the original 72.82.

4.2 Final results

After having analyzed the improvements discussed in Section 4.1, most were updated in UDConverter. The only improvement not added was including *obl:arg* as a possible dependency relation. The difference between *obl* and *obl:arg* is semantic, and it is not accounted for in IcePaHC sentences. It therefore proved complicated to add the relation to the converter, and external information would have to be obtained in order for *obl:arg* to be used.

The four other types of errors discussed above were improved, resulting in error rates shown in Table 1. Rules regarding heads of end-of-sentence punctuation were improved, and the resulting error rate is 29.03%. Rules on head selection of copular verbs were improved by examining individual errors, which resulted in a 7.99% error rate. Head selection of the *cc* dependency relation was also improved, again by examining individual occurrences and adding to the converter's rules. This resulted in a 3.70% error rate. The final improvement made to UDConverter was to the *acl* dependency relation. As discussed in Section 4.1, this error was simple to correct, and rules in the converter were updated to account for this, resulting in a 31.25% error rate.

As discussed, these improvements resulted in the current LAS of 82.87 and UAS of 87.91. These accuracy scores are not consistent with the projected LAS of 85.34, which assumes that all error instances are handled and that the *obl:arg* dependency relation is added to the converter. Nevertheless, the error rates drop considerably, the LAS increasing by 10.05 points and the UAS by 7.12 points. These accuracy scores were obtained by

Deprel to fix	Prev. error rate	Final error rate
punct	75.63%	29.03%
cop	21.86%	7.99%
cc	18.52%	3.70%
acl	72.01%	31.25%
obl:arg	27.73%	27.73%

Table 1: Dependency relations associated with errors in the converter along with the converter's possible LAS after being improved, with respective score gain.

measuring on the same test set as the one that was used for initial evaluation. This method presents some limitations and can cause a bias in the results. The improvements to the converter might be overfitted on the test set, resulting in higher accuracy scores. To counteract this, a development set must be created, manually correcting more sentences and using them to obtain updated accuracy scores.

5 Comparison

Various automatic methods are available to create a UD corpus for Icelandic. To determine the most beneficial method of creating Icelandic UD corpora, we compare UDConverter's accuracy scores to three UD parsers: a UDPipe 1 (Straka and Straková, 2017) model specifically trained to be compared to UDConverter, and the two parsers discussed in Section 2; the Diaparser-based one and the COMBO-based one.

Our UDPipe model was trained on the converted UD_Icelandic-IcePaHC and was used to parse the same sentences as the manually corrected parliament speeches, which were then compared to our manual corrections. While the model tags correctly 92.87% of the time using the Universal Dependencies tagset (UPOS) and 86.78% of the time with the IcePaHC tagset (XPOS), the LAS is only 55.29 and UAS 63.03, which is substantially lower than the output of our converter. Using the same test set, we measured the accuracy of the Diaparser-based parser and the COMBO-based parser. Diaparser delivers a 71.46 LAS and a 78.29 UAS, and the COMBO-based one delivers a 71.04 LAS and a 77.71 UAS. These accuracy scores, in comparison to the scores for UDConverter, are shown in Table 2.

All three parsers, which are the only available Icelandic UD parsers, are trained using output from the converter, which presents some limitations when comparing them to the converter. The parsers learn from the training data, and can never produce results which are as accurate as the data itself. Comparing the parsers' output to the converter's output is therefore not an equal comparison, but it does give an idea about their accuracy. Furthermore, accuracy scores for UDConverter are possibly higher than if they were obtained from development data, as discussed above. Current scores show that using UDConverter to create UD corpora will deliver the most accurate results, as the highest accuracy score for the three parsers is 11.41 points less than the converter's accuracy. However, each method has its advantages and drawbacks, as a converter requires a treebank which is annotated in the appropriate format, while parsers can create a corpus from plain text.

Method	LAS	UAS
UDPipe	55.29	63.03
Diaparser	71.46	78.29
Combo-parser	71.04	77.71
UDConverter	82.87	87.91

Table 2: Accuracy scores of the parsers as compared to UDConverter.

6 Conclusion

We have described the evaluation of a rule-based conversion tool, UDConverter, which converts treebanks in the phrase-structured PPCHE format to the dependency-based UD format. Converted texts were manually corrected and used as testing data. We focused on the accuracy of dependency heads and dependency relations to achieve labeled and unlabeled accuracy scores (LAS, UAS), which serve as F1 scores in our evaluation.

Our results show that UDConverter achieves an LAS of 82.87 and a UAS of 87.91. We compared these accuracy results to accuracy scores of three different Icelandic UD parsers, our UDPipe model along with Diaparser and Combo-parser, which showed that using UDConverter most accurately delivers an Icelandic UD corpus.

Acknowledgements

This project was funded by the Language Technology Programme for Icelandic 2019–2023. The programme, which is managed and coordinated by Almannarómur (https://almannaromur.is/), is funded by the Icelandic Ministry of Education, Science and

Culture. We would like to thank the anonymous reviewers for their contribution.

References

Þórunn Arnardóttir, Hinrik Hafsteinsson, Einar Freyr Sigurðsson, Kristín Bjarnadóttir, Anton Karl Ingason, Hildur Jónsdóttir, and Steinþór Steingrímsson. 2020. A Universal Dependencies Conversion Pipeline for a Penn-format Constituency Treebank. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 16–25, Barcelona, Spain (Online).

Giuseppe Attardi, Daniele Sartiano, and Maria Simi. 2021. Biaffine Dependency and Semantic Graph Parsing for Enhanced Universal Dependencies. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 184–188, Online. Association for Computational Linguistics.

Jón Friðrik Daðason and Hrafn Loftsson. 2022. Pretraining and Evaluating Transformer-based Language Models for Icelandic. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7386–7391, Marseille, France. European Language Resources Association.

Timothy Dozat and Christopher D. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*, Toulon, France.

Atli Jasonarson, Steinþór Steingrímsson, Einar Freyr Sigurðsson, and Jón Friðrik Daðason. 2022a. Biaffine-based UD Parser for Icelandic 22.12. CLARIN-IS.

Atli Jasonarson, Steinþór Steingrímsson, Einar Freyr Sigurðsson, and Jón Friðrik Daðason. 2022b. COMBO-based UD Parser for Icelandic 22.12. CLARIN-IS.

Hildur Jónsdóttir and Anton Karl Ingason. 2020. Creating a Parallel Icelandic Dependency Treebank from Raw Text to Universal Dependencies. In *Proceedings of The 12th Language Resources and Evaluation Conference (LREC 2020)*, pages 2924–2931, Marseille, France.

Mateusz Klimaszewski and Alina Wróblewska. 2021. COMBO: A new module for EUD parsing. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 158–166, Online. Association for Computational Linguistics.

Anthony S. Kroch, Beatrice Santorini, and Lauren Delfs. 2004. Penn-Helsinki Parsed Corpus of Early Modern English. CD-ROM. First Edition. Size: 1.8 million words.

Anthony S. Kroch and Ann Taylor. 2000. Penn-Helsinki Parsed Corpus of Middle English. CD-ROM. Second edition. Size: 1.3 million words.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. Dependency Parsing. *Synthesis Lectures on Human Language Technologies*, 2(1):1–127.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Eiríkur Rögnvaldsson, Anton Karl Ingason, Einar Freyr Sigurðsson, and Joel Wallenberg. 2012. The Icelandic Parsed Historical Corpus (IcePaHC). In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012)*, pages 1977–1984, Istanbul, Turkey. European Language Resource Association.

Kristján Rúnarsson and Einar Freyr Sigurðsson. 2020. Parsing Icelandic Alþingi Transcripts: Parliamentary Speeches as a Genre. In *Proceedings of the Second ParlaCLARIN Workshop*, pages 44–50, Marseille, France.

Steinþór Steingrímsson, Sigrún Helgadóttir, Eiríkur Rögnvaldsson, Starkaður Barkarson, and Jón Guðnason. 2018. Risamálheild: A Very Large Icelandic Text Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 4361–4366, Miyazaki, Japan. European Language Resources Association.

Milan Straka and Jana Straková. 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Ann Taylor, Mitchell Marcus, and Beatrice Santorini. 2003. *The Penn Treebank: An Overview*, pages 5–22. Springer Netherlands.

Joel C. Wallenberg, Anton Karl Ingason, Einar Freyr Sigurðsson, and Eiríkur Rögnvaldsson. 2011. Icelandic Parsed Historical Corpus (IcePaHC) 0.9. CLARIN-IS.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Salih Furkan Akkurt, Gabrielė Aleksandravičiūtė, Ika Alfina, Avner Algom, Chiara Alzetta, Erik Andersen, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Glyd Aranes, Maria Jesus Aranzabe, Bilge Nas Arıcan, Þórunn

Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Katla Ásgeirsdóttir, Deniz Baran Aslan, Cengiz Asmazoğlu, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Juan Belieni, Kepa Bengoetxea, Yifat Ben Moshe, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Lauren Cassidy, Maria Clara Castro, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Liyanage Chamila, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Daniela Corbetta, Marine Courtin, Mihaela Cristescu, Philemon Daniel, Elizabeth Davidson, Leonel Figueiredo de Alencar, Mathieu Dehouck, Martina de Laurentiis, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Christian Ebert, Hanne Eckhoff, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Federica Favero, Jannatul Ferdaousi, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Federica Gamba, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Takahiro Harada, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Marivel Huerta Mendez, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Artan Islamaj, Kaoru Ito, Siratun Jannat, Tomáš Jelínek, Apoorva Jha, Katharine Jiang, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Ritván Karahóǧa, Boris Katz, Tolga Kayadelen, Sarveswaran Kengatharaiyer, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Köse, Alexey Koshevoy, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sandra Kübler, Adrian Kuqi, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phng Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yixuan Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Krister Lindén, Nikola Ljubešić, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Ilya Makarchuk, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Stella Markantonatou, Héctor Martínez Alonso, Lorena Martín Rodríguez, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Tatiana Merzhevich, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Lng Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Hulda Óladóttir, Adédayọ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Noam Ordan, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Teresa Paccosi, Alessio Palmero Aprosio, Anastasia Panova, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Giulia Pedonese, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Andrea Peverelli, Jason Phelan, Jussi Piitulainen, Rodrigo Pintucci, Tommi A Pirinen, Emily Pitler, Magdalena Plamada, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Robert Pugh, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Mizanur Rahoman, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja,

Siva Reddy, Mathilde Regnault, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eiríkur Rögnvaldsson, Ivan Roksandic, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Ben Rozonoyer, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Marta Sartor, Mitsuya Sasaki, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Maria Shvedova, Janine Siewert, Einar Freyr Sigurðsson, João Ricardo Silva, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Haukur Barri Símonarson, Kiril Simov, Dmitri Sitchinava, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Barbara Sonnenhauser, Shafi Sourov, Carolyn Spadine, Rachele Sprugnoli, Vivian Stamou, Steinþór Steingrímsson, Antonio Stella, Abishek Stephen, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Daniel Swanson, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Mirko Tavoni, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Sara Tonelli, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sveinbjörn Þórðarson, Vilhjálmur Þorsteinsson, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Elena Vagnoni, Sowmya Vajjala, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Uliana Vedenina, Giulia Venturi, Eric Villemonte de la Clergerie, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Shira Wigderson, Sri Hartati Wijono, Vanessa Berwanger Wille, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Anna Zhuravleva, and Rayan Ziane. 2022. Universal Dependencies 2.11. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Automatic Transcription for Estonian Children's Speech

Agnes Luhtaru[α] **Rauno Jaaska**[α] **Karl Kruusamäe**[τ] **Mark Fishel**[α]
[α]: Institute of Computer Science, University of Tartu, Estonia
[τ]: Institute of Technology, University of Tartu, Estonia
{agnes.luhtaru, rauno.jaaska, karl.kruusamae, mark.fisel}@ut.ee

Abstract

We evaluate the impact of recent improvements in Automatic Speech Recognition (ASR) on transcribing Estonian children's speech. Our research focuses on fine-tuning large ASR models with a 10-hour Estonian children's speech dataset to create accurate transcriptions. Our results show that large pre-trained models hold great potential when fine-tuned first with a more substantial Estonian adult speech corpus and then further trained with children's speech.

1 Introduction

Automatic Speech Recognition (ASR) continues to face challenges in accurately transcribing children's speech. Research efforts are underway to adapt adult ASR models to better handle the unique pronunciation variations and limited vocabulary that are characteristic of children's speech (Thienpondt and Demuynck, 2022; Dutta et al., 2022). These adaptations are necessary due to the limitations of current ASR systems, which often lack adequate representation of children's speech and struggle to generalize to new examples.

Recent advancements in ASR technology, including the use of large transformer-based models and unsupervised pre-training techniques, have resulted in improved performance for adult speech recognition, with the ability to train on a diverse range of data without human annotations (Baevski et al., 2020; Radford et al., 2022; Hsu et al., 2021). These models demonstrate greater robustness and generalization compared to previous systems. However, the effectiveness of these advanced ASR models for children's speech, especially in low-resource languages like Estonian, remains untested.

In this paper, we are investigating two multilingual speech models - Facebook's Wav2Vec2-XLS-R (Babu et al., 2021) and OpenAI's Whisper (Radford et al., 2022) - as potential starting points for building an ASR system transcribing Estonian children's speech. Our objective is to determine the potential of these models in creating low-effort ASR systems for children speaking a low-resource language like Estonian, for which there are no ASR systems for children's speech.

To accomplish this, we fine-tune the XLS-R and Whisper models from scratch using children's speech data. We also fine-tune pre-existing models for the Estonian language with additional children's speech recordings. Furthermore, we compare the quality of the ASR system by evaluating a pre-made Estonian ASR system provided by Microsoft Azure and exploring its fine-tuning capabilities.

Our research indicates that XLS-R models and Whisper models can serve as effective starting points for building an ASR system using only 10 hours of children's speech. However, for optimal performance, these models should first be fine-tuned with Estonian adult speech. We achieve the best word error rate of around 15 using an XLS-R model that was fine-tuned with Estonian ASR datasets and further trained with children's speech. Furthermore, our results show that the Azure speech-to-text model performs similarly to the Estonian XLS-R and Whisper models but not as well as the fine-tuned public models. Two models that achieved the lowest WER scores are available in HuggingFace[1][2].

In the next sections, we describe which data we used for evaluation and training, which models we used and how we fine-tuned these and last but not

[1] https://huggingface.co/tartuNLP/xls-r-300m-et-children
[2] https://huggingface.co/tartuNLP/whisper-large-v2-et-children

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 705–709

least we present and analyse the results.

2 Dataset and evaluation

The Children ASR dataset used in this work consists of speech recordings from 53 children aged 6 to 13. The data was collected by the Children's Clinic of Tartu University Hospital and contains a mix of both boys and girls speaking about various topics such as answering questions, describing pictures, talking about their family and friends, and more. The dataset is divided into three subsets - test, dev, and train - with no overlap in speakers or texts.

The test set contains all age and gender groups and has a total recording duration of 278 minutes (approximately 4.6 hours). The development set is missing some speakers and has a total recording duration of 182 minutes (approximately 3 hours). The training set is also missing some speakers and has a total recording duration of 613 minutes (approximately 10 hours). A breakdown of the total recording duration for the test set by age and gender of the speakers is shown in Table 1.

Age	Girls (min)	Boys (min)	Total (min)
6	17	21	38
7	14	16	30
8	17	14	31
9	22	18	40
10	15	17	32
11	20	17	37
12	16	22	38
13	19	13	32
Total	140	138	278

Table 1: Total recording duration in minutes for the Estonian children ASR test set, broken down by age and gender of the speakers.

The children in the dataset speak about a wide range of topics, covering everything from answering questions and describing pictures to discussing their family and friends. They also include recordings of children reading fairytales, reciting poems, and saying specific sentences. The utterances in the dataset vary in their level of spontaneity - some are unscripted expressions of thoughts, while others feature children reading.

We evaluate the performance of our speech recognition models using the standard measure of word error rate (WER). This involves converting all text to lowercase and removing punctuation but not standardizing different spelling variations. Our reference transcriptions reflect the pronunciation of children, including any errors they may make. However, the line between correct and incorrect pronunciation is often blurry and some children's speech can be difficult to comprehend. We do not consider the ambiguity in human transcriptions and simply compare the models' output to our reference transcription, which could lead to increased WERs.

3 Models and training

We are using both public large speech models and private black box speech service. In the case of public models, we also searched for models already fine-tuned with Estonian speech data. We fine-tune the selection of these models with the children's speech dataset mentioned in the last section.

For public models, we use two multilingual ones: Facebook's XLS-R and OpenAI's Whisper (Radford et al., 2022). XLS-R model is trained with speech modelling objective, not ASR but it can be fine-tuned to ASR with Connectionist Temporal Classification (CTC) (Graves et al., 2006) algorithm. The Whisper on the other hand is a multipurpose model that contains both transformer encoder and decoder blocks and has been trained on several speech-processing tasks, like multilingual speech recognition, speech translation and voice activity detection (Radford et al., 2022).

The available XLS-R models have 300 million, 1 billion and 2 billion parameters, we are using the two smaller ones in this work. The Whisper model comes in six different sizes; we are using medium and large-v2 since the Estonian error rates for other ones are relatively high. There is one Estonian-specific fine-tuned model available for the 300 million parameter version, trained with over 700 hours of Estonian speech data (Alumäe and Olev, 2022). There are several Estonian Whisper models available in HuggingFace but these are trained with fewer data examples. We are using the best available medium and large-v2 ones.[3][4]. Following the submission of this paper, a new Estonian Whisper model was released[5], which is

[3] https://huggingface.co/agnesluhtaru/whisper-medium-et-ERR2020

[4] https://huggingface.co/agnesluhtaru/whisper-large-et-ERR2020-v2

[5] https://huggingface.co/TalTechNLP/whisper-medium-et

trained using a larger dataset. In the scope of this work, we evaluate the model but do not fine-tune it using children's speech.

We use standard fine-tuning procedures. For training XLS-R-based ASR models from scratch, we use the learning rate of 3e-4, a 400-step warmup and train the models for 60 epochs with children's speech dataset, which is less than 4000 steps. When further fine-tuning the Estonian XLS-R model with children's speech, we use the learning rate of 2e-5 and 200 warmup steps. We fine-tune all the Whisper models with warmup 10% of the steps and learning rate 1e-05. When fine-tuning the out-of-the-box Whisper models, we train these for 5000 steps or atound 40 epochs and when fine-tuned models already trained with Estonian adult speech, we train the large model for 2000 steps or over 16 epochs and medium model for 1000 steps or eight epochs.

For the private model, we use Microsoft Azure Speech service's speech-to-text[6], which requires an Azure subscription and a Speech resource. The transcription services can be accessed by making REST requests.

Microsoft Azure offers the option to fine-tune the model with custom datasets. This process involves uploading data to train the models, followed by deploying the trained models. Since audio-based fine-tuning is not available for Estonian, we use text-based tuning for our work with the texts from the children's speech dataset.

4 Results

In this section, we describe the results of all the models based on Facebook's XLS-R, OpenAI'S Whisper and Microsoft Azure speech-to-text.

4.1 XLS-R

Table 2 shows the word error rate (WER) scores of fine-tuned Estonian XLS-R models using only 10 hours of Estonian children's speech data, the fine-tuned Estonian model (Alumäe and Olev, 2022) and Estonian model further trained with children's speech. We can see that the limited amount of data for fine-tuning XLS-R from scratch results in a high WER of over 30 for both models with 300 million and one billion parameters. Training an ASR model using only 10 hours of speech data can be challenging, especially when the speech is for a low-resource language and children.

[6] https://learn.microsoft.com/en-us/azure/cognitive-services/speech-service/speech-to-text

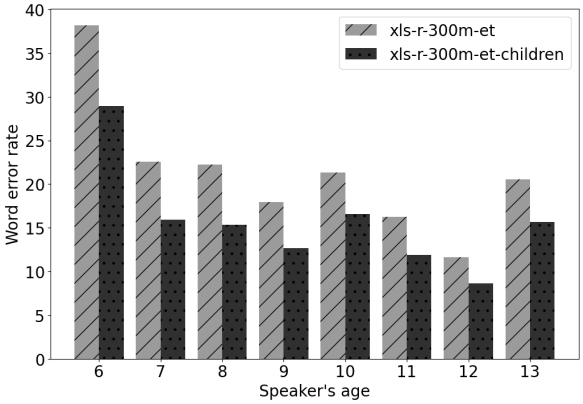


Figure 1: Performance comparison of Estonian XLS-R ASR and children's speech fine-tuned models across age groups.

Model	Dev	Test
xls-r-300M-children	34.58	36.3
xls-r-1B-children	31.06	30.89
xls-r-300M-et	19.15	20.62
xls-r-300M-et-children	14.30	15.31

Table 2: Comparison of WER scores for Facebook's Wav2Vec2 XLS-R (Babu et al., 2021) based models fine-tuned with only Estonian children's speech, only Estonian adult speech (Alumäe and Olev, 2022) and first fine-tuned to Estonian and further trained with children's speech.

The results show that the pre-trained Estonian ASR model has a WER of around 20, while further fine-tuning the model with children's speech data leads to even better results, with a WER of less than 15. Based on the lower WER score for fine-tuned one billion parameter model, we can suggest that a larger model fine-tuned with Estonian data first and then further trained on children's speech could lead to even better results.

The results indicate that fine-tuning the Estonian ASR model using children's speech data improves performance across all age groups (refer to Figure 1). Younger speakers tend to have a higher word error rate (WER) than older speakers, although this relationship is not always straightforward. There are some exceptions, such as the recognition performance for 13-year-olds being worse than that of younger age groups. This high-

lights that speaker variability plays a role in the WER results. Nevertheless, the fine-tuning of the ASR model using children's speech data reduces the differences in recognition performance across age groups, resulting in improved overall performance.

4.2 Whisper

The performance of the out-of-the-box Whisper models on the children's dataset (see Table 3) is comparable to the scores reported by Radford et al. (2022) on the Estonian Common Voice 9 Ardila et al. (2020). All models have a WER of at least 35. So, although we can use Whisper without fine-tuning, it does not transcribe Estonian speech well and therefore does not give great transcriptions for Estonian children's speech as well.

When fine-tuning the model using only 10 hours of children's speech, we can already achieve better results. The large-v2 model yields a WER of around 20, which is significantly better than some models fine-tuned with Estonian speech alone. The medium model, developed by TalTechNLP and trained with over 800 hours of Estonian speech[7], outperforms the XLS-R model that was trained solely on Estonian adult speech.

Model	Dev	Test
Whisper-medium	43.21	46.11
Whisper-large-v2	35.06	36.01
Whisper-medium-children	24.29	25.08
Whisper-large-v2-children	20.58	20.38
TalTech Whisper-medium-et	15.64	17.26
Whisper-medium-et	26.83	28.78
Whisper-large-v2-et	28.13	29.2
Whisper-medium-et-children	17.49	18.66
Whisper-large-v2-et-children	15.73	16.02

Table 3: Comparison of WER scores for OpenAI Whisper (Radford et al., 2022) models and Whisper models fine-tuned with only Estonian children's speech, only Estonian adult speech and first fine-tuned to Estonian and further trained with children's speech.

Despite using the Estonian Whisper models fine-tuned with fewer audio text pairs than the XLS-R model, when trained further with children's speech, the large model achieved similar WER as the double fine-tuned smaller XLS-R model. The difference between TalTechNLP's whisper-medium-et and whisper-large-v2-et-children is small, suggesting that fine-tuning the former with children's data could potentially result in even better performance.

4.3 Azure

The results from our evaluation of the children's speech dataset show that the out-of-the-box Azure speech-to-text model performs similarly or better than the fine-tuned Estonian XLS-R model (Alumäe and Olev, 2022) but worse than Estonian Whisper medium trained by TalTechNLP. As indicated in Table 4, the Microsoft Azure speech-to-text scores are around 20 or below.

Model	Dev	Test
Microsoft Azure	20.18	18.93
Azure text-tuned	21.21	20.31

Table 4: WER scores for Microsoft Azure speech-to text and its custom text-tuned version.

However, the experiment also shows that text-tuning is not the best approach for this particular dataset. The dataset mostly contains simpler vocabulary and not much terminology, most likely leading to quick overfitting with text-tuning. Currently, text-tuning is the only option available for the Estonian language, but it might not be the best use case for children's speech datasets.

5 Discussion

Our experiments show that children's speech recognition continues to be a tricky problem but big speech models are looking promising. It is possible to build an ASR system for Estonian children's speech without any bells and whistles using only 10 hours of data and get output that is decent and might be good enough for use in chatbots. However, when it comes to six-year-olds, whose speech is difficult to understand even for the human ear, the system is still struggling.

We evaluate different models and it appears that both OpenAI's Whisper and Facebook's XLS-R are viable options for developing a speech recognition model for Estonian children's speech. The current best word error rate is around 15 with XLS-R. However, it remains unclear if this pre-trained model is optimal for children's speech or if a lower error rate could be achieved with Whisper after fine-tuning with a similar amount of Estonian

[7] https://huggingface.co/TalTechNLP/whisper-medium-et

adult speech. Additionally, we do not obtain comparable results with the Azure service, as it does not permit fine-tuning with audio data.

Our findings suggest that the results could be improved by using a larger XLS-R model as the base or by fine-tuning Whisper models with more data. Additionally, we do not use a separate language model, which is possible with both Whisper and XLS-R models and could potentially enhance the performance of these models.

6 Conclusion

We test the performance of two speech recognition models, XLS-R and Whisper, on transcribing Estonian children's speech. We fine-tune the models with children's speech data and compared them to an off-the-shelf system from Microsoft Azure. Both models fine-tuned with children's speech, outperform Microsoft Azure, which does not allow fine-tuning with audio for Estonian, and are promising for children's ASR system.

Acknowledgements

This research has been in part supported by European Social Fund via IT Academy programme, Estonian Centre of Excellence in IT (EXCITE) funded by the European Regional Development Fund, and AI & Robotics Estonia co-funded by the EU and Ministry of Economic Affairs and Communications in Estonia.

References

Tanel Alumäe and Aivo Olev. 2022. Estonian speech recognition and transcription editing service. volume 10, page 409–421.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. https://aclanthology.org/2020.lrec-1.520 Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. https://doi.org/10.48550/ARXIV.2111.09296 Xls-r: Self-supervised cross-lingual speech representation learning at scale.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Satwik Dutta, Sarah Anne Tao, Jacob C. Reyna, Rebecca Elizabeth Hacker, Dwight W. Irvin, Jay F. Buzhardt, and John H.L. Hansen. 2022. https://doi.org/10.21437/Interspeech.2022-555 Challenges remain in Building ASR for Spontaneous Preschool Children Speech in Naturalistic Educational Environments. In *Proc. Interspeech 2022*, pages 4322–4326.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. https://doi.org/10.1145/1143844.1143891 Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA. Association for Computing Machinery.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. https://doi.org/10.1109/TASLP.2021.3122291 Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 29:3451–3460.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. https://doi.org/10.48550/ARXIV.2212.04356 Robust speech recognition via large-scale weak supervision.

Jenthe Thienpondt and Kris Demuynck. 2022. https://doi.org/10.21437/Interspeech.2022-10964 Transfer Learning for Robust Low-Resource Children's Speech ASR with Transformers and Source-Filter Warping. In *Proc. Interspeech 2022*, pages 2213–2217.

Translated Benchmarks Can Be Misleading: the Case of Estonian Question Answering

Hele-Andra Kuulmets **Mark Fishel**
Institute of Computer Science
University of Tartu
{hele-andra.kuulmets, mark.fisel}@ut.ee

Abstract

Translated test datasets are a popular and cheaper alternative to native test datasets. However, one of the properties of translated data is the existence of cultural knowledge unfamiliar to the target language speakers. This can make translated test datasets differ significantly from native target datasets. As a result, we might inaccurately estimate the performance of the models in the target language. In this paper, we use both native and translated Estonian QA datasets to study this topic more closely. We discover that relying on the translated test dataset results in overestimation of the model's performance on native Estonian data.

1 Introduction

Translating test datasets to the target language has become a popular alternative to creating datasets from scratch in the target language (Yang et al., 2019, Ponti et al., 2020, Conneau et al., 2018). The main reason for this is that translating data, either manually or automatically, and reannotating it is easier than hiring data annotators to annotate the data. In addition, to ensure the quality of the newly created dataset, the authors often go through an exhaustive process of verifying the data quality, making creating new datasets even more expensive. On the other hand, existing datasets are already established in the NLP community. Another benefit of translated datasets is that they make evaluating cross-lingual transfer learning easier, as the identical datasets make the results directly comparable across languages.

However, in case only a translated test dataset exists for a specific task in a specific language, it is also likely true that there is probably no task-specific native training data available in that language. If there was native training data available, then a small subset of it could have been used to create a test dataset. Creating only a training dataset with no target test dataset available would also provide no benefit to the creators.

The existence of (translated) test dataset in some specific language, together with the non-existence of training data in the same language, has created an interesting situation where translated datasets have been mostly employed to advance cross-lingual transfer learning or related methods (e.g. TRANSLATE-TEST).[1] However, this contradicts the idea of these methods, which is to generalize to languages where training data for the task is unavailable. With translated test datasets, the training data *is* usually available[2]; it is just in another (source) language. In fact, it is most likely used to train the model, which will be evaluated with the translated test dataset. Because of this, there is a danger that evaluation results become artificially inflated and overestimate the model's performance on native data.

This paper aims to study the concerns of using translated test datasets more closely. We use English as a source language and Estonian as a target language and evaluate models trained on the source language with native and translated target datasets to see how the results on translated dataset compare to the results on the native dataset. We opt for TRANSLATE-TEST setup because it can be generalized more easily to different tasks as only a model trained in English is needed. In addition, it is competitive or even better at solving Estonian language understanding tasks than cross-lingual transfer methods (see Table 1).

[1] Some translated datasets, e.g XQuAD (Artetxe et al., 2020b) are specifically created to advance cross-lingual transfer research. Although the purpose of translating the dataset may differ, the outcome has the same issues that are addressed in this paper.

[2] Only test or validation split is usually translated.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 710–716

Dataset	Task	Metric	TRTE	TRTR	CL	Native	SOTA
EstQA (Käver, 2021)	extractive QA	F1	73.0	**79.9**	73.4	49.20	82.4
News Stories (Härm and Alumäe, 2022)	abstractive summarization	ROUGE-1	**17.22**	17.0	-	16.22	TRTE
XCOPA ET (Ponti et al., 2020)	commonsense reasoning	accuracy	**81.0**	57.4†	79.0‡	-	TRTE/81.0‡

Table 1: Comparison of different methods on solving Estonian language understanding tasks. **TRTE**: TRANSLATE-TEST; **TRTR**: TRANSLATE-TRAIN; **CL**: cross-lingual transfer learning; **Native**: only native data was used for training; **SOTA**: reported state-of-the-art in literature (arbitrary method). Results are reported by the authors of the datasets if not specified otherwise: † Ruder et al. (2021); ‡ Muennighoff et al. (2022).

2 Related Work

TRANSLATE-TEST and TRANSLATE-TRAIN are commonly used machine translation baselines for cross-lingual transfer learning studies. (Conneau et al., 2018, Ponti et al., 2020, Lin et al., 2022, Hu et al., 2020, Liu et al., 2019). Somewhat surprisingly, TRANSLATE-TEST has shown to be a superior method for many languages in a cross-lingual setting where target language training data is not available (Ponti et al., 2020, Lin et al., 2022). Meanwhile, TRANSLATE-TRAIN has also been shown to outperform cross-lingual transfer learning methods and can compete with TRANSLATE-TEST (Ruder et al., 2021).

The success of machine translation-based methods has motivated researchers to improve these methods even more. Yu et al. (2022) shows that TRANSLATE-TRAIN can be improved by learning a mapping from originals to translationese that is applied during test time to the originals of the target language. Dutta Chowdhury et al. (2022) employs a bias-removal technique to remove translationese signals from the classifier. Oh et al. (2022) proposes TRANSLATE-ALL - a method that uses both techniques simultaneously. Their model is trained both on data in the source language and source data translated to the target language. During inference, the two predictions, one on the target dataset and another on the target dataset translated to the source language, are ensembled. Isbister et al. (2021) shows that even if a training dataset is available in the target language, it might still be beneficial to translate both training and test datasets to English to employ pre-trained English language models instead of native language models. Artetxe et al. (2020a) draws attention to the fact that even human-translated datasets can contain artifacts that can hurt the performance of the model when compared to the native English datasets. He shows that the performance drop is indeed caused by the fact that training is done on the original data while testing is done on translated data.

3 Methodology

Our goal is to compare evaluation results obtained with native and translated Estonian question-answering datasets in a TRANSLATE-TEST setting where the data is machine translated to English and fed to a model also trained on English. We hypothesize that translated test dataset will overestimate results on the native test dataset.

3.1 Models

XLM-RoBERTa (Conneau et al., 2020) A multilingual encoder trained on 100 languages (including Estonian) with masked language modeling objective. We fine-tune the base model XLM-ROBERTA-BASE.[3]

3.2 Datasets

SQuAD (Rajpurkar et al., 2016) An English extractive question-answering dataset consisting of more than 100 000 crowdsourced question-answer-paragraph triplets. The paragraphs are from English Wikipedia.

XQuAD (Artetxe et al., 2020b) A cross-lingual extractive question-answering benchmark that consists of 1190 triplets from SQuAD's validation set translated to 10 languages (not including Estonian) by professional translators. Each question has exactly one correct answer.

[3]https://huggingface.co/xlm-roberta-base

EstQA (Käver, 2021) An Estonian extractive question-answering dataset consisting of 776 train triplets and 603 test triplets where each question in the test dataset has possibly more than one correct answer. The paragraphs are from Estonian Wikipedia. It was specifically created to be an Estonian equivalent for English SQuAD.

3.3 XQuAD$_{et}$

We also need a translated Estonian question-answering dataset to see whether our hypothesis is true. This dataset should ideally be created using the same methodology as was used for the native dataset EstQA to avoid a situation where the difference in results could be attributed to different methodologies. Since EstQA was created by following the methodology used for SQuAD and XQuAD is a subset of it, we decided to translate the English subset of XQuAD to Estonian. The translation was done with Google Cloud API. The annotation spans were first automatically aligned with SimAlign (Jalili Sabet et al., 2020). After that, the alignments were verified manually, and corrections were made if necessary. We denote this dataset as **XQuAD$_{et}$**. Similarly to XQuAD, it consists of 1190 triplets.

3.4 Training and Inference

We train our QA model by fine-tuning XLM-ROBERTA-BASE SQuAD dataset. Ideally, we would have used existing QA models as this is one of the main benefits of the TRANSLATE-TEST approach. However, since XQuAD is a subset of the validation set of SQUAD, then this would have given an unfair advantage to XQuAD in our experiments.

During inference, the input (in Estonian) is machine translated to English using Google Cloud API and fed to a model trained on SQUAD. The predicted span (in English) is then automatically aligned with the input in Estonian using SimAlign to project the prediction back to Estonian.

3.5 Evaluation

Following Rajpurkar et al. (2016) we evaluate our models with exact match (EM) and f1 score (F1). Exact match is a metric that measures the percentage of predictions that match any of the gold labels exactly while F1 measures the average overlap between the predicted and gold answer. We use the official scoring script of SQuAD.[4]

Train data	Test data	EM	F1
SQuAD	XQuAD$_{et}$	**58.74**	**72.26**
	EstQA	57.04	70.35

Table 2: TRANSLATE-TEST results on Estonian QA datasets.

Train data	Test data	EM	F1
EstQA	EstQA$_{en}$	**26.37**	41.99
	XQuAD	24.21	**43.64**

Table 3: TRANSLATE-TEST results on English QA datasets.

4 Results

Table 2 summarizes the main results of our experiments. The results support our hypothesis that using translated test datasets together with TRANSLATE-TEST can lead to overestimating the performance on the native target data. Note that in order to obtain the predictions for **XQuAD$_{et}$** the data was machine translated twice (first to Estonian and then during the inference back to English) but is still more easily solvable, despite the potentially stacking translation errors that can diminish the meaning of the texts.

4.1 Symmetry Test

We conducted an additional experiment to see whether our hypothesis is also true in the opposite direction, i.e., the model is trained on Estonian data and English test data is translated to Estonian during the inference. For that purpose, EstQA was translated to English using the same pipeline as for **XQuAD$_{et}$**. However, the results shown in Table 3 do not provide clear evidence that our hypothesis is also true in the opposite direction. Additionally, it can be seen that the results on both datasets are very low, which is expected since the EstQA training dataset contains only 776 training samples.

4.2 Quality of Automatic Annotations

The pipeline of solving QA task with TRANSLATE-TEST consists of multiple components, all of which work with some error rate. We can not assess the quality of machine-translated datasets because we do not have gold translations. However, both **XQuAD$_{et}$** and

[4]More precisely, we use `evaluate` library that wraps the original scripts: https://github.com/huggingface/evaluate.

Dataset	EM	F1
EstQA$_{en}$	64.30	83.67
XQuAD$_{et}$	83.61	91.40

Table 4: Annotation quality of automatic annotations.

EstQA$_{en}$ contain human-verified annotations which we can compare against automatically obtained annotations. Table 4 shows the quality of automatic alignments on translated test datasets as measured with EM and F1 against manually corrected annotations. As the table shows, automatic alignments were much better for translated XQuAD, especially when comparing EM scores with nearly 20% difference.

The aligner algorithm in all our experiments was IterMax from the SimAlign package with a distortion of 0.5, as suggested by the authors. We used embeddings from BERT-BASE-MULTILINGUAL-CASED[5] (Devlin et al., 2019) as this yielded the best results in our experiments when compared to other contextual embeddings (see Appendix A for more details).

5 Discussion

5.1 Machine vs Human Translated Datasets

One may argue that in order to show that translated datasets are inferior to native datasets, human-translated data should be used instead of machine-translated data because usually translated datasets are created with the help of professional translators. However, we believe that it is not necessary. Firstly, it has been shown that regardless of the method, translated data contains translationese which makes it different from native data (Volansky et al., 2013, Bizzoni et al., 2020). Secondly, the cultural knowledge incorporated into the translated datasets will make them differ from native data despite the translation method. Finally, our goal was to investigate whether the model's performance would be overestimated with translated test datasets. Intuitively, this is more difficult to show with machine-translated data because of potential translation errors. Therefore, if the hypothesis is true with machine-translated data, it is fair to assume that it will also be true for human-translated data.

[5]https://huggingface.co/bert-base-multilingual-cased

5.2 Cause of Mismatch

The problem we are addressing in this paper is caused by the fact that data from the same distribution is often used to train and evaluate models in a TRANSLATE-TEST setting where cultural differences of languages should naturally be taken into account. However, one may say that this argumentation leads to the same conclusions about monolingual research because it also uses different splits of the same dataset for training and testing. Although domain shift is a problem in monolingual research, it differs from the scenario addressed in this paper. Domain mismatch happens because the model learns to detect unwanted biases in the training dataset that are irrelevant to solving the task in general (McCoy et al., 2019, Jia and Liang, 2017). The mismatch in our scenario happens because different cultural knowledge is naturally intertwined into each of the languages by the speakers, which the model trained only on one language can not know about.

5.3 Asymmetry

Our experiments showed that overestimating happens when native Estonian data is translated to English but not when native English data is translated to Estonian during test-time data augmentation, i.e., not always are translated datasets easier to solve for the model. However, the results might also be affected by the properties of the underlying language model or train dataset size. For a more fair comparison of translation directions, the train datasets should be around the same size. Currently, the difference in sizes is more than 100 times.

5.4 Limitations

The main limitation of the paper is its relatively small scale which can be overcome by including more languages, more datasets, or a cross-lingual transfer scenario. Alternatively, one can translate test datasets from languages other than English to Estonian (or any other target language) and compare the performance in TRANSLATE-TEST (et $\rightarrow$ en) setup.

6 Conclusion

We compared the performance of an English extractive QA model on native and translated Estonian test datasets in TRANSLATE-TEST setting to see how results on the translated dataset compare

to the results on the native dataset. Our experiments showed that results on the translated dataset overestimate the results on the native dataset.

Acknowledgements

This article has been financed/supported by European Social Fund via "ICT programme" measure.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2020a. https://doi.org/10.18653/v1/2020.emnlp-main.618 Translation artifacts in cross-lingual transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7674–7684, Online. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020b. https://doi.org/10.18653/v1/2020.acl-main.421 On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Yuri Bizzoni, Tom S Juzek, Cristina España-Bonet, Koel Dutta Chowdhury, Josef van Genabith, and Elke Teich. 2020. https://doi.org/10.18653/v1/2020.iwslt-1.34 How human is machine translationese? comparing human and machine translations of text and speech. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 280–290, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. https://doi.org/10.18653/v1/2020.acl-main.747 Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. https://doi.org/10.18653/v1/D18-1269 XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. https://doi.org/10.18653/v1/N19-1423 BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Koel Dutta Chowdhury, Rricha Jalota, Cristina España-Bonet, and Josef Genabith. 2022. https://doi.org/10.18653/v1/2022.naacl-main.292 Towards debiasing translation artifacts. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3983–3991, Seattle, United States. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. https://proceedings.mlr.press/v119/hu20b.html XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Henry Härm and Tanel Alumäe. 2022. Abstractive summarization of broadcast news stories for estonian. In *Proceedings of Baltic HLT 2022*, page 511–524, Riga, Latvia. Baltic Journal of Modern Computing.

Tim Isbister, Fredrik Carlsson, and Magnus Sahlgren. 2021. https://aclanthology.org/2021.nodalida-main.42 Should we stop training more monolingual models, and simply use machine translation instead? In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 385–390, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. https://doi.org/10.18653/v1/2020.findings-emnlp.147 SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. https://doi.org/10.18653/v1/D17-1215 Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Anu Käver. 2021. Extractive question answering for estonian language. Master's thesis, Tallinn University of Technology.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Jiahua Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2019. https://doi.org/10.18653/v1/P19-1227 XQA: A cross-lingual open-domain question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2358–2368, Florence, Italy. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. https://doi.org/10.18653/v1/P19-1334 Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2022. https://doi.org/10.48550/ARXIV.2211.01786 Crosslingual generalization through multitask finetuning.

Jaehoon Oh, Jongwoo Ko, and Se-Young Yun. 2022. Synergy with translation artifacts for training and inference in multilingual tasks. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. https://doi.org/10.18653/v1/2020.emnlp-main.185 XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. https://doi.org/10.18653/v1/D16-1264 SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021. https://doi.org/10.18653/v1/2021.emnlp-main.802 XTREME-R: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vered Volansky, Noam Ordan, and Shuly Wintner. 2013. https://doi.org/10.1093/llc/fqt031 On the features of translationese. *Digital Scholarship in the Humanities*, 30(1):98–118.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. https://doi.org/10.18653/v1/D19-1382 PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Sicheng Yu, Qianru Sun, Hao Zhang, and Jing Jiang. 2022. https://doi.org/10.18653/v1/2022.acl-short.40 Translate-train embracing translationese artifacts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 362–370, Dublin, Ireland. Association for Computational Linguistics.

A Performance of SimAlign with different embeddings

Since the authors of SimAlign did not evaluate their choice of embedding on Estonian, we did our own evaluation with three different embeddings. Figure 1 and Figure 2 show how the choice of embedding affects the quality of alignments.

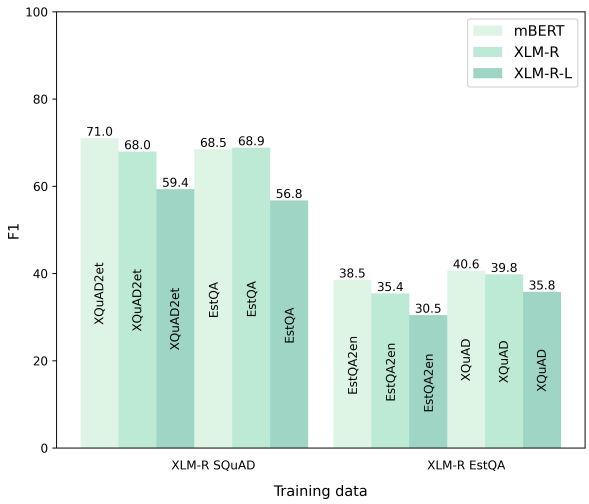


Figure 1: F1 of automatically aligned answers with different embeddings.

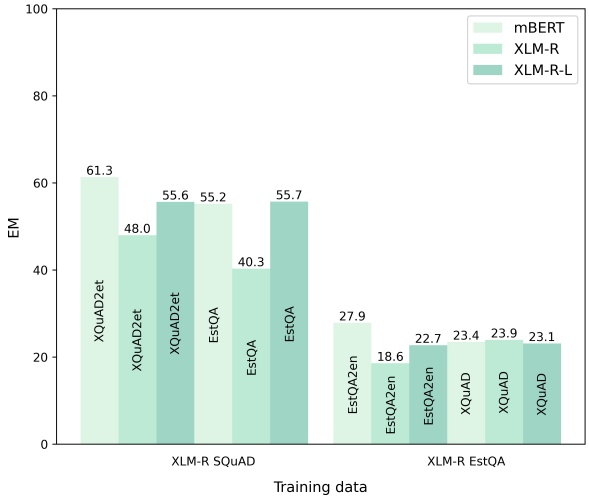


Figure 2: EM of automatically aligned answers with different embeddings.

The scores are obtained by comparing predictions projected back to the target language with gold annotations. As the authors of SimAlign, we found that embeddings from mBERT produce the best alignments. Note that the scores obtained with mBERT are not the same as shown in Table 2. This is because the algorithm that projected predicted spans back to the target language was slightly changed before obtaining the final results.

B Hyperparameters

For both English and Estonian QA models, XLM-R was fine-tuned with learning rate $2e^{-5}$ (linear decay) and batch size 16 for 20 epochs with early stopping after ten consecutive evaluation steps with no improvement in validation loss. The model was evaluated after every 100 steps. Weight decay was 0.01, warmup ratio 0.

Predicting the presence of inline citations in academic text using binary classification

Peter Vajdečka
Prague University of Economics and Business
Prague, Czechia
vajp02@vse.cz

Elena Callegari
University of Iceland
Reykjavík, Iceland
ecallegari@hi.is

Desara Xhura
SageWrite ehf.
Reykjavík, Iceland
dxhura@gmail.com

Atli Snær Ásmundsson
University of Iceland
Reykjavík,
Iceland
atliasmunds@gmail.com

Abstract

Properly citing sources is a crucial component of any good-quality academic paper. The goal of this study was to determine what kind of accuracy we could reach in predicting whether or not a sentence should contain an inline citation using a simple binary classification model. To that end, we fine-tuned SciBERT on both an imbalanced and a balanced dataset containing sentences with and without inline citations. We achieved an overall accuracy of over 0.92, suggesting that language patterns alone could be used to predict where inline citations should appear.

1 Introduction

Providing accurate, relevant citations is an essential part of academic writing. Not only do citations allow authors to better contextualize the results of the paper, but they also lend credibility and authority to the claims made in the article. Failing to give credit to existing research when credit is due, on the other hand, is taken to show a lack of academic integrity, and is strongly frowned upon by the academic community. Appropriately adding citations, however, is not trivial: even humans sometimes struggle to determine where inline citations should go, and what should or should not be cited. This is particularly true in the case of junior academics and students (Vardi, 2012) (Carson et al., 1992) (Pennycook, 1996). In the context of *automatic* text evaluation, determining where citations should go is even less straightforward. One way in which one could automatically determine whether a given paragraph requires (additional) inline citations is through automatic plagiarism detection systems. However, processing a document to determine whether some sections of it have been plagiarized can require a considerable amount of time, particularly if the document exceeds a certain length. Building a plagiarism checker is also complicated, as the process requires scanning the full web for documents, and possibly obtaining access to research articles that might lay behind a paywall. Finally, results might not always be accurate (Kohl Kerstin, 2012), as the checker might fail in finding similarities between concepts simply because sentences that are identical in meaning have been expressed through a different formulation. Because of these downsides, we were interested in exploring how much mileage we could get out of a simple binary classification experiment trying to predict whether or not a given sentence should include an inline citation. In particular, we reasoned that it should be possible to predict at least to some extent whether a sentence should contain an inline citation simply by looking at the presence vs. absence of specific lexical cues. For example, verbs such as "claimed", nouns such as "authors" and phrases such as "as seen in" tend to appear together or in the vicinity of inline citations. The same holds true of some capitalized nouns (e.g. "Attention", "Minimalism").

1.1 Related Work

References play an essential role in academia and as such have been the focus of several NLP studies (Iqbal et al., 2021). Some of the properties that researchers have traditionally focused on are extracting the polarity of inline citations (is the referenced article negatively or positively mentioned?) (Abu-Jbara et al., 2013), and determining the purpose of inline citations (Viswanathan et al., 2021). Our paper builds on a body of research that has attempted to predict the "citation worthiness" (Cohan et al., 2019) of sentences, i.e. whether or not a given sentence should contain an inline citation. Several approaches have been suggested to determine the citation worthiness of text, see in particular (Beel et al., 2016), (Färber and Jatowt,

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 717–722

2020) and (Ma et al., 2020) for an overview. We have also seen an increased tendency towards using references as a way to build knowledge graphs (Viswanathan et al., 2021) and speed up the search for relevant research articles. There is also a tendency towards using references to aid automated text summarization (Yasunaga et al., 2019).

1.2 Motivation

Developing shallow automated techniques that can detect whether or not a sentence should contain an inline citation has several practical applications. A shallow inline-citation predictor can be used to (i) help academics identify forgotten inline citations, i.e. citations that the author meant to add at the review stage but ultimately forgot to include, (ii) guide junior researchers in the paper-writing process, flagging concepts or ideas that might require attribution, (iii) improving the coverage of automatic essay analyzers, and (iv) in the context of natural language generation, decreasing the chances of committing plagiarism by flagging passages that might require a citation.

2 Preparing the Data

To determine what types of inline citation styles are used in different research disciplines, we randomly selected two articles for each of the following 18 research fields: Medicine, Biology, Chemistry, Engineering, Computer Science, Physics, Math, Psychology, Economics, Political Science, Business, Geology, Sociology, Geography, Environmental Science, Art, History, Philosophy. After analyzing these 36 articles, we concluded that most of the articles adopted the IEEE, APA or the Chicago reference styles.

We first created an initial dataset consisting of 2000 research articles; these were randomly selected from the ArXiv and PubMed datasets (Cohan et al., 2018) that are freely available on the Huggingface Datasets library (Lhoest et al., 2021) (`https://huggingface.co/datasets/scientific_papers`).

These 2000 articles were subsequently processed to discard articles with a citation pattern other than the IEEE, APA or Chicago reference styles. The pre-processing task of detecting inline citations was handled through a simple Python script. Using regular expressions, different kinds of citation styles were mapped to corresponding regex capture patterns. We started by writing regexes that would match the three citation styles that we identified as the most frequently used: IEEE, APA and Chicago. Later on, we also decided to include the alpha BibTeX style, as that appears to be quite frequently used in ArXiV papers. The Python script did the following: first, every given citation pattern was extracted from the article's plain text. Then, the style with the highest capture count was set as the article's default style. This means that even when the extraction process found inline citations that matched a style that was not the article's primary citation style, the script was still able to identify the primary style. Finally, the inline citations matching the primary style were substituted with an -ADD-CITATION- token; this step is important as it allowed us to generalize across different referencing styles. If for some reason no citation style was detected, the token replacement failed, and the article was discarded from further analysis.

We then created a second dataset by taking all the articles with IEEE, APA or Chicago as reference styles and by (i) breaking down the original text into sentences, and assigning each sentence to a separate entry, (ii) assigning different labels to entries containing inline citations and entries not containing inline citations, and (iii) removing the -ADD-CITATION- token throughout the dataset. This second dataset features 411'992 sentences (entries), of which 54'735 contain an inline citation (see Table 1). The dataset is accessible at `https://github.com/elenaSage/InlineCitationSet` and is free to use. This second dataset is the dataset we used for the classification experiments that we describe below.

No Citation	Contains Citation	Total
357257	54735	411992

Table 1: Composition of Inline Citation Dataset

3 Classification model

In our research, we intend to train a classification model that can determine whether a sentence should contain a citation (positive class) or not (negative class) depending on the text input. In the first column of the Table 2, an example of the input text is displayed. The model we aim to train for this input text should predict that a citation must be present; this is a positive class prediction. If the

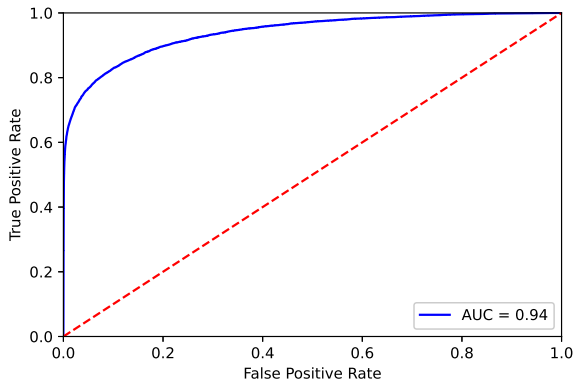


Figure 1: ROC curve on testing imbalanced dataset

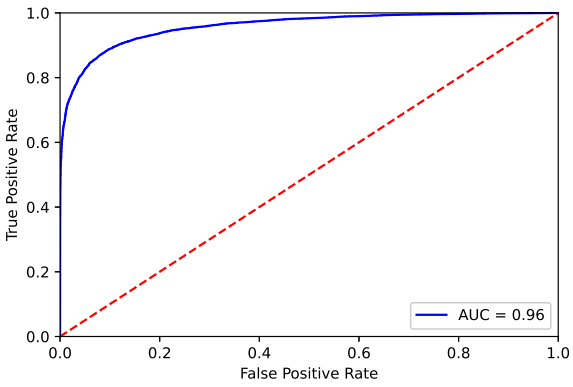


Figure 2: ROC curve on testing balanced dataset

model predicts a negative class, it would mean that the text should not contain any citation.

In recent years, BERT-based language models (Devlin et al., 2019) have achieved state-of-the-art performance in numerous NLP classification tasks. Due to their pre-training on massive corpora and fine-tuning for a specific downstream purpose, these models can acquire accurate language representations.

Our Inline Citation dataset includes scientific data containing science-specific terminology. Because of that, we decided to encode texts for the classification task using the BERT architecture that has been pre-trained on scientific texts, i.e. the SciBERT model (Beltagy et al., 2019). Exactly like BERT, SciBERT contains 30K word-piece tokens, but unlike BERT its vocabulary is pertinent to the scientific area. In the scientific domain, SciBERT outperforms BERT in a variety of tasks (Beltagy et al., 2019) and achieves SOTA performance in multi-class text classification on the SciCite dataset (Cohan et al., 2019). It has been demonstrated that fine-tuned uncased SciBERT with SciVocab followed by a linear layer produces the best results for scientific data (Beltagy et al., 2019) or for citation context classification (Maheshwari et al., 2021). Therefore we use this model in each experiment.

4 Fine-tuning SciBERT

Research papers generally contain more sentences without inline citations than sentences with citations, which leads to having more examples for the ”no citation” class. Performing classification tasks using imbalanced datasets poses multiple challenges, the most prominent being the bias towards the most represented class (He and Garcia, 2009). There are multiple studies that try to counteract this phenomenon by bringing more balance in the distribution of classes within the same dataset (see for example (Mohammed et al., 2020) or (Krawczyk, 2016)). Two well-known techniques in direction of balancing are undersampling and oversampling. Undersampling however also presents drawbacks, the most important one being the loss of information that might be captured by the most represented class. With this in mind, we decided to run classification experiments on both the full (imbalanced) dataset and a more balanced subset of the dataset which we obtained by undersampling the data. We divided both the balanced and the imbalanced dataset into a training subset (60%), a validation subset (20%) and a test subset (20%), resulting in a ”60:20:20” split. The split was then modified so that the proportion of positive (sentences containing a citation) to negative (sentences not containing a citation) texts in each subset would not be altered following the split (see Table 3).

Next, we fine-tuned all SciBERT parameters end-to-end utilizing the training and validation subsets. For fine-tuning, we adhered primarily to the similar design and optimization decisions utilized in articles (Beltagy et al., 2019; Devlin et al., 2019). We used the ReLu activation function in linear one-layer feed-forward classifier which inputs the last hidden state of the [CLS] token. In other words, this last hidden state of the [CLS] token is utilized as the sequence’s features to feed the classifier.

We experimented with numerous hyper-

Table 2: An illustration of text input and prediction output

Input sentence	Class
In particular, our colored pebbles generalize and strengthen the previous results of Lee and Streinu and give a new proof of the Tutte-Nash-Williams characteri- zation of arboricity.	Positive
The tidal friction theories explain that the present rate of tidal dissipation is anomalously high because the tidal force is close to a resonance in the response function of ocean.	Positive
A k-map-graph is a graph that admits a decomposition into k edge-disjoint map-graphs.	Negative

Dataset type	Class	Training subset	Validation subset	Testing subset
Balanced	Contains citation	32831	10957	10947
	No citation	36085	12015	12025
Imbalanced	Contains citation	32739	11049	10947
	No citation	214455	71350	71452

Table 3: Dataset split

parameters for fine-tuning with both datasets. We fine-tuned for 2 to 5 epochs using batch size 16, 32 or 50 and learning rate of 5e-5, 5e-6, 1e-5 or 2e-5, with a dropout of 0.1 or without dropout. We optimized cross-entropy loss with the assistance of the AdamW optimizer (Kingma and Ba, 2014). The best results were obtained when the models were fine-tuned for 2 epochs with a batch size of 50 samples and a learning rate of 5e-5 without dropout, followed by a linear warmup and linear decay (Devlin et al., 2019); this was the case for both the balanced and the imbalanced dataset. We used softmax to determine probabilities for predictions, with a threshold of 0.7 proving optimal, meaning that sentences with a calculated probability greater than 0.7 are predicted to be positive, i.e. they are predicted to contain an inline citation.

5 Discussion

In our work, we always consider positive labels as a class of those input texts that contain an inline citation. This means that we always understand True-Positives (TP) as correctly predicted texts that contain an inline citation (see graph 1 and graph 2). This is also analogous to the Precision and Recall calculations and the derived F-score in graphs 3 and 4, and the metrics in Table 4 below. The focus is mainly on this class of inline citations as positive, since it is definitely a minority with respect to quantity, which makes the problem more challenging.

We report the results of our two experiments in Table 4. We see that balancing the dataset by undersampling helped to significantly reduce the bias towards the most represented class, increasing the recall of the least represented class (=sentences containing an inline citation) from 0.63 to 0.84.

Since we used both balanced and imbalanced datasets, useful performance indicators include the Area under the Curve AUC for the precision-recall curve PR or the Receiver Operating Characteristic curve ROC (Bradley, 1997; Hanley and McNeil, 1982). Figure 1 and figure 2 reveal that the ROC curves are nearly comparable in both datasets, with the imbalanced dataset having a slightly lower AUC value of 0.94 against that of 0.96 for the balanced dataset. For imbalanced data, however, a PR plot is advised (Sun et al., 2009; Gu et al., 2009); our PR plots are depicted in figure 3 and 4. The imbalanced dataset's PR curve follows a different path than the balanced dataset's PR curve, which is also reflected in its considerably lower AUC value (=0.84) compared to that of the balanced dataset (=0.96).

6 Conclusion

The goal of this paper was to determine how effective binary classification models can be at predicting whether or not sentences appearing in academic articles should contain an inline citation.

Citation prediction				
Approach	**Precision**	**Recall**	**F1 score**	**Accuracy**
Balanced SciBERT validation	0.93	0.84	0.89	0.90
Balanced SciBERT testing	0.93	0.84	0.88	0.89
Imbalanced SciBERT validation	0.92	0.63	0.75	0.94
Imbalanced SciBERT testing	0.92	0.64	0.75	0.94

Table 4: Prediction results

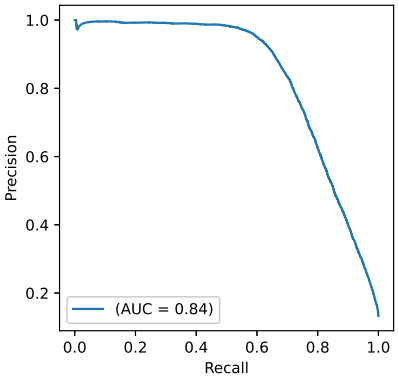


Figure 3: PR curve on testing imbalanced dataset

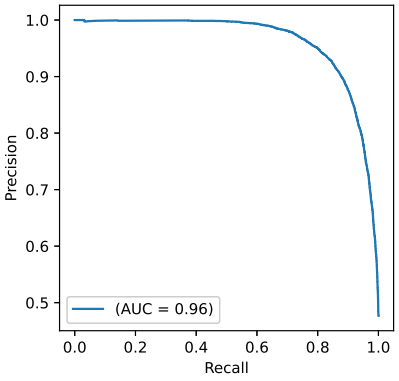


Figure 4: PR curve on testing balanced dataset

To that end, we used regular expressions to identify inline citations in published research papers, and then created a dataset composed of 411k sentences, where approximately 54k contained inline citations. We then ran a fine-tuned SciBERT classifier on both a balanced and imbalanced dataset, achieving an overall accuracy of over 0.92. This result shows that language patterns alone could be used to predict the presence of inline citations in academic text with a reasonable degree of accuracy. We presented the problem as a binary classification task on the sentence level, i.e. we only considered the target sentence and did not consider the context in which the sentence appeared, for example by also looking at the sentences appearing before and after the target sentence. Taking into account the previous and the following sentence could be worthwhile in that some inline citations scope over multiple contiguous sentences rather than just refer to a single sentence (i.e. the concept of "citing area" first mentioned in (Nanba and Okumura, 1999)). The sentences contained in the Inline Citation Dataset however are all sequential: they come in the same sequence as they were found in the original paper. This means that information on the context in which a given target sentence appears is already available in our dataset. This paves the path for further experiments that take contextual sentential information into account, such as using transformers to predict in which position inline citations should appear.

References

Amjad Abu-Jbara, Jefferson Ezra, and Dragomir Radev. 2013. Purpose and polarity of citation: Towards nlp-based bibliometrics. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 596–606.

Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. 2016. Paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17:305–338.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. https://doi.org/10.18653/v1/D19-1371 SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Andrew P Bradley. 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159.

Joan G. Carson, Nancy D. Chase, Sandra U. Gibson, and Marian F. Hargrove. 1992. Literacy demands of the undergraduate curriculum. *Literacy Research and Instruction*, 31(4):25–50.

Arman Cohan, Waleed Ammar, Madeleine Van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. *arXiv preprint arXiv:1904.01608*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. https://doi.org/10.18653/v1/N19-1423 BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Michael Färber and Adam Jatowt. 2020. Citation recommendation: approaches and datasets. *International Journal on Digital Libraries*, 21(4):375–405.

Qiong Gu, Li Zhu, and Zhihua Cai. 2009. Evaluation measures of the classification performance of imbalanced data sets. In *Computational Intelligence and Intelligent Systems: 4th International Symposium, ISICA 2009, Huangshi, China, October 23-25, 2009. Proceedings 4*, pages 461–471. Springer.

James A Hanley and Barbara J McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Haibo He and Edwardo A. Garcia. 2009. https://doi.org/10.1109/TKDE.2008.239 Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

Sehrish Iqbal, Saeed-Ul Hassan, Naif Radi Aljohani, Salem Alelyani, Raheel Nawaz, and Lutz Bornmann. 2021. A decade of in-text citation analysis based on natural language processing and machine learning techniques: An overview of empirical studies. *Scientometrics*, 126(8):6551–6599.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Eleonora Kohl Kerstin. 2012. https://doi.org/10.3402/rlt.v19i3.7611 Fostering academic competence or putting students under general suspicion? voluntary plagiarism check of academic papers by means of a web-based plagiarism detection system. *Research in Learning Technology*, 19.

Bartosz Krawczyk. 2016. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, et al. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184.

Shutian Ma, Chengzhi Zhang, and Xiaozhong Liu. 2020. A review of citation recommendation: from textual content to enriched context. *Scientometrics*, 122:1445–1472.

Himanshu Maheshwari, Bhavyajeet Singh, and Vasudeva Varma. 2021. Scibert sentence representation for citation context classification. In *Proceedings of the Second Workshop on Scholarly Document Processing*, pages 130–133.

Roweida Mohammed, Jumanah Rawashdeh, and Malak Abdullah. 2020. Machine learning with oversampling and undersampling techniques: overview study and experimental results. In *2020 11th international conference on information and communication systems (ICICS)*, pages 243–248. IEEE.

Hidetsugu Nanba and Manabu Okumura. 1999. Towards multi-paper summarization using reference information. In *IJCAI*, volume 99, pages 926–931.

Alastair Pennycook. 1996. Borrowing others' words: Text, ownership, memory, and plagiarism. *TESOL quarterly*, 30(2):201–230.

Aixin Sun, Ee-Peng Lim, and Ying Liu. 2009. On strategies for imbalanced text classification using svm: A comparative study. *Decision Support Systems*, 48(1):191–201.

Iris Vardi. 2012. Developing students' referencing skills: a matter of plagiarism, punishment and morality or of learning to write critically? *Higher Education Research Development*, 31(6):921–930.

Vijay Viswanathan, Graham Neubig, and Pengfei Liu. 2021. Citationie: Leveraging the citation graph for scientific information extraction.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander Fabbri, Irene Li, Dan Friedman, and Dragomir Radev. 2019. ScisummNet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of AAAI 2019*.

Neural Text-to-Speech Synthesis for Võro

Liisa Rätsep **Mark Fishel**
Institute of Computer Science
University of Tartu
{liisa.ratsep, mark.fisel}@ut.ee

Abstract

This paper presents the first high-quality neural text-to-speech (TTS) system for Võro, a minority language spoken in Southern Estonia. By leveraging existing Estonian TTS models and datasets, we analyze whether common low-resource NLP techniques, such as cross-lingual transfer learning from related languages or multi-task learning, can benefit our low-resource use case. Our results show that we can achieve high-quality Võro TTS without transfer learning and that using more diverse training data can even decrease synthesis quality. While these techniques may still be useful in some cases, our work highlights the need for caution when applied in specific low-resource scenarios, and it can provide valuable insights for future low-resource research and efforts in preserving minority languages.

1 Introduction

The advancements in neural text-to-speech (TTS) technology have greatly improved the quality of speech synthesis for many languages. However, despite the potential benefits of TTS for facilitating accessibility and language preservation, developing TTS systems for low-resource languages remains challenging due to the limited availability of training data for these languages.

Võro, a Finno-Ugric minority language spoken in Southern Estonia, serves as a great example of a low-resource language that could benefit from TTS technology. While linguistic resources for Võro are limited, the language is closely related to Estonian – a high-resource Finno-Ugric language with significantly more datasets, tools, and pretrained models.

The goal of this paper is to present the first high-quality neural TTS system for Võro and evaluate various low-resource NLP techniques for improving synthesis quality for the language. By leveraging existing Estonian TTS models and datasets, we investigate the impact of transfer learning from related languages and multi-speaker and multilingual approaches on the TTS quality of Võro.

The main contributions of this paper are:

1. We develop the first high-quality neural text-to-speech system for Võro and make it publicly available[1].

2. We show that having only 1.5 hours of Võro speech data per speaker is sufficient to develop TTS systems for low-resource languages without using cross-lingual transfer learning or additional monolingual data.

3. We highlight the potential negative effects of diversifying low-resource TTS datasets with data from closely related languages.

2 Background

As neural text-to-speech models require vast amounts of data, existing research has proposed several approaches to mitigate the issue of insufficient training data. For example, several works have shown that cross-lingual pretraining improves the quality of low-resource TTS systems (Chen et al., 2019; Xu et al., 2020).

In a survey on multilingual strategies for low-resource TTS, Do et al. (2021) evaluated the usefulness of using multilingual datasets for improving low-resource language performance. They observed that for sequence-to-sequence models, including additional data from other languages is almost always beneficial and often overweighs the negative effect of having a lower ratio of target data in the entire training dataset. The authors also noted that there is no clear evidence that

[1] https://neurokone.ee

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 723–727

using supporting languages from the same language family is more beneficial but claimed that using a shared input representation space (such as phonemes) may be more important.

At the same time, using closely related languages to boost low-resource performance has been successfully used for many text-based NLP tasks, including for developing Finno-Ugric machine translation systems that also include the Võro language (Tars et al., 2021). Unfortunately, the usage of neural methods for Võro has so far been limited to this example. There is also no existing research on Võru TTS. While the Estonian Language Institute and the Võro Institute have collaborated to create an HMM-based TTS system for Võro[2], this work has not been described in research.

3 Methodology

In this section, we present our methodology and experiment setup. Our approach evaluates the benefits of low-resource TTS approaches when training non-autoregressive Transformer-based models (Ren et al., 2019; Łańcucki, 2021). We focus on three common strategies – cross-lingual transfer learning from a pre-trained Estonian TTS model, combining data from multiple Võro speakers, and including Estonian data to create a multilingual system. Additionally, we explore data augmentation to handle the orthographic variation of Võro.

3.1 Datasets

Our experiments used speech data from two Võro speakers – an adult male and a child (female). Both datasets were attained from the Estonian Language Institute and contained an identical set of 1132 sentences, out of which 100 were set aside for evaluation purposes.

The Estonian dataset consisted of 6 male and 4 female speakers from the Speech Corpus of Estonian News Sentences (Fishel et al., 2020) and the Estonian Language Institute's audiobook corpora (Piits, 2022a,b). A subset of 1000 sentences per speaker was selected from the Estonian corpora to balance the training dataset.

The audio files were resampled at 22050 Hz and converted into mel-spectrograms using a Hann window with a frame size of 1024 and a hop length of 256. The mel-spectrogram frames were aligned to the graphemes using the Estonian alignment model by Alumäe et al. (2018). Training a separate alignment model for Võro was also considered, but initial testing showed that the Estonian model was successfully able to produce high-quality alignments. The alignment was also used to trim excessive pauses in the audio.

All datasets were lowercased, and punctuation was normalized to a limited set of characters to reduce the vocabulary size. In total, the training dataset contained 3 hours of Võro and 14 hours of Estonian speech.

3.2 Data Augmentation

While the Võro dataset follows a standardized version of Võro orthography, many speakers and well-known news outlets do not conform to this standard. For example, the glottal stop (*q*) may be omitted or used only when it affects the meaning of the word, and some speakers may also use an apostrophe instead the letter *q*. Similarly, an apostrophe or an acute accent that marks palatalization is often used only when it affects the meaning.

In order to create a system that could successfully synthesize speech from all common written formats of Võro, we considered this to be an important challenge. As there are no existing NLP tools for Võro that would allow us to analyze these features automatically, we decided to use data augmentation to generate orthographic alternatives where glottal stops or palatalization features were removed for the system to cope with different orthographies.

Additionally, while our dataset contained the letter *y*, all cases of it were replaced with *õ* as they are no longer differentiated according to the orthographic standardization changes from 2005.

3.3 Model Configuration

All models were trained using an open-source implementation[3] of a non-autoregressive Transformer-based (Vaswani et al., 2017) model. The architecture is similar to FastPitch (Łańcucki, 2021) with explicit duration and pitch prediction components. An existing multispeaker model for Estonian (Rätsep et al., 2022) was used for our cross-lingual transfer learning experiments. In multispeaker systems, the speaker identity was marked with a prepended global style token (Wang et al., 2018).

[2] https://www.eki.ee/~indrek/voru/index.php

[3] https://github.com/TartuNLP/TransformerTTS

We trained models with three different data configurations – single-speaker Võro models for each speaker, multi-speaker Võro models with both speakers, and multi-speaker multilingual models with both Estonian and Võro data. For each data configuration, we also trained another model, which was initialized using the weights of the existing Estonian model. All models were trained for at least 400k steps and using identical hyperparameters.

4 Results

To assess the quality of the models, we conducted a mean opinion score (MOS) (Chu and Peng, 2001) evaluation[4] among volunteers from the Võro community. The evaluators were required to know the Võro language but did not have to be native speakers. Of the 41 volunteers, 6 considered themselves native speakers, and 9 had a self-reported Võru level of C1 or higher. Many participants with lower levels of Võru knowledge also mentioned that their passive language skills were higher as they mostly used Võro when communicating with older family members who were native speakers.

The evaluation used a subset of 50 random sentences per speaker (100 total per method) from the held-out dataset, and the samples were generated using pretrained HiFiGAN (Kong et al., 2020) models. The appropriate model for each speaker was selected by evaluating samples generated with multiple vocoder models. For the lower-pitched male speaker, we used a model trained on the VCTK dataset (Yamagishi et al., 2019), and for the child speaker, we used a model trained on the LJ Speech (Ito and Johnson, 2017) corpus and finetuned on Tacotron 2 (Shen et al., 2018) output. We also included ground truth samples from the held-out dataset and ground truth samples converter to mel-spectrograms and reconstructed by the same vocoder models.

The evaluation results can be seen in Table 1. Expectedly, ground truth samples in their original and reconstructed forms scored the highest among the participants. From the TTS models, the highest scores were given to single-speaker models. These were followed by the multi-speaker Võro models, but the performance drop from the single-speaker models should not be considered significant. The multilingual models showed consistently worse performance compared to the monolingual models. Additionally, we observe minor benefits from using cross-lingual transfer learning.

Method	MOS
Ground truth	4.03 ± 0.12
Ground truth + vocoder	3.83 ± 0.13
Single-speaker	3.55 ± 0.15
Single-speaker (transfer)	3.62 ± 0.15
Multi-speaker	3.43 ± 0.15
Multi-speaker (transfer)	3.50 ± 0.13
Multilingual	3.10 ± 0.15
Multilingual (transfer)	3.29 ± 0.15

Table 1: Mean opinion scores with 95% confidence intervals on the held-out dataset.

In addition to scoring samples, participants were encouraged to comment on their overall impressions of speech quality and the evaluation process. Many expressed a positive surprise about synthesis quality and mentioned the presence of TTS artifacts, such as crackling, as their main evaluation criteria. Some participants also noted that while almost all samples were intelligible, they did not always sound like a native Võro speaker, especially when producing the glottal stop sound. Unfortunately, as the participants did not know which models produced which samples, further analysis would be needed to assess whether all models are equally prone to this issue and whether it can also be observed in ground truth examples.

5 Discussion and Future Work

Unexpectedly, our MOS evaluation results are in conflict with existing low-resource TTS literature that reports benefits from diversifying training data with samples from other speakers or related languages and from using cross-lingual transfer learning. This brings into question both the usefulness of these techniques as well as our approach.

Firstly, it could be argued that the observations about the low negative performance impact of data imbalance by Do et al. (2021) may not apply to non-autoregressive Transformer-based systems, as the study focused on other methods, such as recurrent or convolutional neural networks. Therefore, the performance drop in multilingual models could still be caused by an imbalance between the

[4] https://tartunlp.github.io/TransformerTTS/nodalida2023/

two languages in the dataset. Alternatively, as our model size was dictated by the existing pretrained Estonian models, it may lack sufficient capacity to work in a multilingual setting.

Additionally, it is possible that we should no longer consider Võro a low-resource language in this task. Based on initial testing with Estonian datasets, we found that the required amount of speech data for Transformer-based models to produce coherent speech is between 1-2 hours, and improvements from using more data are significantly less noticeable. Similar observations about reduced data requirements for Transformer-based models have also been recently reported by Pine et al. (2022). In our case, we had 1.5 hours of speech per speaker, and it may have been sufficient for us not to benefit from additional data from other speakers. Alternatively, as the two Võro datasets contained identical sentences, they may not differ sufficiently to benefit from each other. However, a more detailed evaluation methodology could be considered to measure the effects on specific features of synthetic speech, such as prosodic variability or pronunciation mistakes.

As our work focused on creating a high-quality system for Võro without applying artificial constraints, such as using smaller subsets of the high-resource datasets, these points were not explicitly explored in our work. However, in the future, low-resource TTS strategies should be further reviewed specifically for Transformer-based architectures and for different levels of resource constraint. Until then, these strategies should be used with caution and evaluated for each specific low-resource scenario.

6 Conclusion

This article presented the first high-quality neural text-to-speech system for the Võro language. We explored the usage of Estonian TTS models and datasets to boost the performance of our low-resource use case.

Our results suggest that we can achieve high-quality Võro TTS without transfer learning or using data from multiple speakers or closely related languages. While these techniques may still be helpful in some cases, we highlight the need for further research and evaluation when applied in specific low-resource scenarios.

References

Tanel Alumäe, Ottokar Tilk, and Asadullah. 2018. Advanced rich transcription system for Estonian speech. In *Human Language Technologies - the Baltic Perspective: Proceedings of the Eighth International Conference*, pages 1–8. IOS Press.

Yuan-Jui Chen, Tao Tu, Cheng chieh Yeh, and Hung-Yi Lee. 2019. End-to-End Text-to-Speech for Low-Resource Languages by Cross-Lingual Transfer Learning. In *Proc. Interspeech 2019*, pages 2075–2079.

Min Chu and Hu Peng. 2001. An objective measure for estimating MOS of synthesized speech. In *EUROSPEECH 2001, 7th European Conference on Speech Communication*, pages 2087–2090. ISCA.

Phat Do, Matt Coler, Jelske Dijkstra, and Esther Klabbers. 2021. A Systematic Review and Analysis of Multilingual Data Strategies in Text-to-Speech for Low-Resource Languages. In *Proc. Interspeech 2021*, pages 16–20.

Mark Fishel, Annika Laumets-Tättar, and Liisa Rätsep. 2020. Speech corpus of Estonian news sentences. https://doi.org/10.15155/9-00-0000-0000-0000-001ABL.

Keith Ito and Linda Johnson. 2017. The LJ Speech dataset. https://keithito.com/LJ-Speech-Dataset/.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems*, pages 17022–17033. Curran Associates, Inc.

Adrian Łańcucki. 2021. FastPitch: Parallel text-to-speech with pitch prediction. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6588–6592.

Liisi Piits. 2022a. Estonian female voice audiobook corpus for speech synthesis. https://doi.org/10.15155/3-00-0000-0000-0000-090D4L.

Liisi Piits. 2022b. Estonian male voice audiobook corpus for speech synthesis. https://doi.org/10.15155/3-00-0000-0000-0000-08BF4L.

Aidan Pine, Dan Wells, Nathan Brinklow, Patrick Littell, and Korin Richmond. 2022. Requirements and motivations of low-resource speech synthesis for language revitalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland. Association for Computational Linguistics.

Liisa Rätsep, Rasmus Lellep, and Mark Fishel. 2022. Estonian text-to-speech synthesis with non-autoregressive transformers. *Baltic Journal of Modern Computing*, 10.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. FastSpeech: Fast, robust and controllable text to speech. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. 2018. Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783.

Maali Tars, Andre Tättar, and Mark Fišel. 2021. Extremely low-resource machine translation for closely related languages. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 41–52, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A. Saurous. 2018. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5180–5189. PMLR.

Jin Xu, Xu Tan, Yi Ren, Tao Qin, Jian Li, Sheng Zhao, and Tie-Yan Liu. 2020. Lrspeech: Extremely low-resource speech synthesis and recognition. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 2802–2812, New York, NY, USA. Association for Computing Machinery.

Junichi Yamagishi, Cristophe Veaux, and Kirsten MacDonald. 2019. CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92). `https://datashare.ed.ac.uk/handle/10283/3443`.

Transfer to a Low-Resource Language via Close Relatives: The Case Study on Faroese

Vésteinn Snæbjarnarson[1,2] **Annika Simonsen**[3] **Goran Glavaš**[4] **Ivan Vulić**[5]
[1]University of Copenhagen [2]Miðeind ehf [3]University of Iceland
[4]University of Würzburg [5]University of Cambridge

Abstract

Multilingual language models have pushed state-of-the-art in cross-lingual NLP transfer. The majority of zero-shot cross-lingual transfer, however, use *one and the same* massively multilingual transformer (e.g., mBERT or XLM-R) to transfer to *all* target languages, irrespective of their typological, etymological, and phylogenetic relations to other languages. In particular, readily available data and models of resource-rich sibling languages are often ignored. In this work, we empirically show, in a case study for Faroese – a low-resource language from a high-resource language family – that by leveraging the phylogenetic information and departing from the 'one-size-fits-all' paradigm, one can improve cross-lingual transfer to low-resource languages. In particular, we leverage abundant resources of other Scandinavian languages (i.e., Danish, Norwegian, Swedish, and Icelandic) for the benefit of Faroese. Our evaluation results show that we can substantially improve the transfer performance to Faroese by exploiting data and models of closely-related high-resource languages. Further, we release a new web corpus of Faroese and Faroese datasets for named entity recognition (NER), semantic text similarity (STS), and new language models trained on all Scandinavian languages.

1 Introduction

Massively multilingual Transformer-based language models (MMTs) such as mBERT (Devlin et al., 2019), XLM-RoBERTa (Conneau et al., 2020a) and mT5 (Xue et al., 2021) have been the driving force of modern multilingual NLP, allowing for rapid bootstrapping of language technology for a wide range of low(er)-resource languages by means of (zero-shot or few-shot) cross-lingual transfer from high(er)-resource languages (Lauscher et al., 2020; Hu et al., 2020; Xu and Murray, 2022; Schmidt et al., 2022). Cross-lingual transfer with MMTs is not without drawbacks. MMTs' representation spaces are heavily skewed in favor of high-resource languages, for which they have been exposed to much more data in pretraining (Joshi et al., 2020; Wu and Dredze, 2020); combined with the 'curse of multilinguality' – i.e., limited per-language representation quality stemming from a limited capacity of the model (Conneau et al., 2020a; Pfeiffer et al., 2022) – this leads to lower representational quality for languages underrepresented in MMTs' pretraining. Cross-lingual transfer with MMTs thus fails exactly in settings in which it is needed the most: for low-resource languages with small digital footprint (Zhao et al., 2021). Despite these proven practical limitations, the vast majority of work on cross-lingual transfer still relies on MMTs due to their appealing conceptual generality: in theory, they support transfer between any two languages seen in their pretraining. Such strict reliance on MMTs effectively ignores the linguistic phylogenetics and fails to directly leverage resources of resource-rich languages that are closely related to a target language of interest.

In this work, we attempt to mitigate the above limitations for a particular group of languages, departing from the 'one-size-fits-all' paradigm based on MMTs. We focus on a frequent and realistic setup in which the target language is a low-resource language but from a high-resource language family, i.e., with closely related resource-rich languages. A recent comprehensive evaluation of the languages used in Europe[1] scores languages

[1]The *Digital Language Equality in Europe by 2030: Strategic Agenda and Roadmap* published by the European Language Equality Programme (ELE), https://european-language-equality.eu/agenda/.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 728–737

based on the available resources. Languages such as German and Spanish score at around 0.5 of the English scores, and more than half of the languages are scored below 0.02 of the English score. Many, including almost all regional and minority languages such as Faroese, Scottish Gaelic, Occitan, Luxembourgish, Romani languages, Sicilian and Meänkieli have the score of (almost) 0. However, what differentiates these languages from low-resource languages from Africa (e.g., Niger-Congo family) or indigenous languages of Latin America (e.g., Tupian family) is the fact that they typically have *closely related high-resource languages as 'language siblings'*. In this case, we believe, language models (LMs) of closely related high-resource languages promise more effective transfer compared to using MMTs, plagued by the 'curse of multilinguality', as the vehicle of transfer.

In this proof-of-concept case study, we focus on Faroese as the target language and demonstrate the benefits of *linguistically informed* transfer. We take advantage of available data and resources from the closely related but much more 'NLP-developed' other Scandinavian languages.[2] We show that using "Scandinavian" LMs brings substantial gains in downstream transfer to Faroese compared to using XLM-R as a widely used off-the-shelf MMT. The gains are particularly pronounced for the task of semantic text similarity (STS), the only high-level semantic task in our evaluation. We further show that adding a limited-size target-language corpus to LM's pretraining corpora brings further gains in downstream transfer. As another contribution of this work, we collect and release: (1) a corpus of web-scraped monolingual Faroese, (2) multiple LMs suitable for Faroese, including those trained on all five Scandinavian languages, and (3) two new task-specific datasets for Faroese labeled by native speakers: for NER and STS.

2 Background and Related Work

Cross-Lingual Transfer Learning with MMTs and Beyond. A common approach to cross-lingual transfer learning involves pretrained MMTs (Devlin et al., 2019; Conneau et al., 2020a; Xue et al., 2021). These models can be further pretrained for specific languages or directly adapted for downstream tasks. A major downside of the MMTs has been dubbed the *curse of multilinguality* (Conneau et al., 2020a), where the model becomes saturated and performance can not be improved further for one language without a sacrifice elsewhere, something which continued pretraining for a given language alleviates (Pfeiffer et al., 2020). Adapter training, such as in (Pfeiffer et al., 2020; Üstün et al., 2022), where small adapter modules are added to pretrained models, has also enabled cost-efficient adaptation of these models. The adapters can then be used to fine-tune for specific languages and tasks without incurring catastrophic forgetting.

Other methods involve translation-based transfer (Hu et al., 2020; Ponti et al., 2021), and transfer from monolingual language models (Artetxe et al., 2020; Gogoulou et al., 2022; Minixhofer et al., 2022). Bilingual lexical induction (BLI) is the method of mapping properties, in particular embeddings, from one language to another via some means such as supervised embedding alignment, unsupervised distribution matching or using an orthogonality constraint (Lample et al., 2018; Søgaard et al., 2018; Patra et al., 2019), and has also been used to build language tools in low-resource languages (Wang et al., 2022).

Attempts to alleviate the abovementioned issues have been made, such as vocabulary extension methods (Pfeiffer et al., 2021), which add missing tokens and their configurations to the embedding matrix. Phylogeny-inspired methods have also been used where adapters have been trained for multiple languages and stacked to align with the language family of the language of interest (Faisal and Anastasopoulos, 2022). Some analysis on the effects of using pretrained MMTs has been done: Fujinuma et al. (2022) conclude that using pretrained MMTs that share script and overlap in the family with the target language is beneficial. However, when adapting the model for a new language, they claim that using as many languages as possible (up to 100) generally yields the best performance.

Inspired by this line of research, in this work, we focus on improving MMT-based cross-lingual transfer for a particular group of languages, those that have sibling languages with more abundant data and resources.

NLP Resources in Scandinavian Languages. A fair amount of language resources have been devel-

[2]The Scandinavian languages are a family of Indo-European languages that form the North Germanic branch of the Germanic languages. The largest languages of the family are: (1) Danish (population 5.8M), Norwegian (5.4M) and Swedish (10.4M) – the Mainland Scandinavian languages, and (2) Icelandic (373K) and Faroese (54K) – the Insular Scandinavian languages.

oped for the Scandinavian languages, particularly if aggregated across all languages of the family. It is also worth mentioning that Danish, Icelandic, Norwegian and Swedish are represented in raw multilingual corpora such as CC100 (Conneau et al., 2020b) or mC4 (Xue et al., 2021) as well as in parallel datasets such as (Schwenk et al., 2021; Agić and Vulić, 2019). Large multilingual language models have been trained on these datasets (Devlin et al., 2019; Liu et al., 2020; Xue et al., 2021) but have been shown to have limited capacity for languages with smaller relative representation in pretraining corpora. Faroese is not included (at least not correctly labelled) in these crawled corpora.This may be in part due to the limited amount of Faroese that can be found online, and in part due to its close relatedness to the other languages of the Scandinavian family (Haas and Derczynski, 2021). A brief overview of prior work in cross-lingual transfer to Faroese is given in Appendix D.

In this work, we use the following open language resources for the Scandinavian languages.

Danish: The Danish Gigaword Corpus (Strømberg-Derczynski et al., 2021) is a billion-word corpus containing a wide variety of text.We also use a NER resource, the DaNE corpus (Hvingelby et al., 2020).

Icelandic: With Icelandic as the most closely related language to Faroese, we experiment with an Icelandic language model, IceBERT (Snæbjarnarson et al., 2022). For the NER experiment, we make use of the MIM-GOLD-NER corpus (Ingólfsdóttir et al., 2020).

Norwegian: The Norwegian Colossal Corpus (NCC) (Kummervold et al., 2022) contains 49GB of clean Norwegian data from a variety of sources, making it the largest such public collection in the Nordics. We also make use of the NorNE (Jørgensen et al., 2020) NER corpus (both for Bokmål and Nynorsk).

Swedish: The Swedish Gigaword Corpus (Eide et al., 2016) contains text from between 1950 and 2015. The latest NER corpus for Swedish is SweNERC (Ahrenberg et al., 2020), where the authors include more modern texts than in earlier corpora.

Faroese: A POS corpus, the Sosiualurin corpus is an annotated Newspaper corpus with 102k words (Hansen et al., 2004). The Faroese Wikipedia has also been used to create a tree bank (Tyers et al., 2018), which has a Universal Dependencies (UD) mapping. We use this corpus along with the FarPaHc (Ingason et al., 2012), which also has a UD mapping.

3 New Faroese Datasets

3.1 Faroese Common Crawl Corpus (FC3)

Faroese monolingual data is scarce, mainly because of the limited size of the Faroese-speaking population. Despite this, we manage to extract a decent amount of varied Faroese text from the Common Crawl corpus (FC3). To this effect, we adopted the approach of Snæbjarnarson et al. (2022) for Icelandic, i.e., we targeted the websites with the Faroese top-level domain (`.fo`). After clean-up and deduplication, the obtained Faroese corpus consists of 98k paragraphs containing in total 9M word-level tokens. Albeit relatively small compared to corpora from other Scandinavian languages, this Faroese corpus still drives significant downstream performance gains (see §5).

3.2 Named Entity Recognition (FoNE)

We annotate the Sosialurin corpus (6,286 lines, 102k words) with named entities following the CoNLL schema using an Icelandic NER-tagger trained using the ScandiBERT model, see §4. The annotation was then manually reviewed. Out of the 118,533 tokens (including punctuation), 9,001 are annotated using the Date (546), Location (1,774), Miscellaneous (332), Money (514), Organization (2,585), Percent (115), Person (2,947) and Time (188) tags. We refer to this new dataset as *FoNE*.

3.3 Semantic Similarity (Fo-STS)

The STS Benchmark (Cer et al., 2017) measures semantic text similarity (STS) between pairs of sentences. For each pair of sentences, the annotators assigned the score (on a Likert 1-5 scale) that indicates the extent to which the two sentences are semantically aligned. We manually translated from English to Faroese 729 sentence pairs from the test portion of the STS Benchmark; the translation was carried out by a native speaker of Faroese fluent in English, who was instructed to preserve in the translation the extent of semantic alignment between the original English sentences.

4 Model Training

We train the following new language models: (i) *ScandiBERT* is trained on concatenated corpora of all Scandinavian languages, (ii) *ScandiBERT-no-fo*

is trained on concatenated corpora of all Scandinavian languages except Faroese (i.e., without any Faroese data, that is, no FC3, Bible or Sosialurin), and (iii) *DanskBERT* which is trained only on the Danish data; we train *DanskBERT* for the purposes of comparison with IceBERT, in the setup in which we carry out downstream transfer to Faroese by means of a monolingual model of a closely related language (with Danish being more distant to Faroese than Icelandic). We additionally evaluate transfer with models that have been further pretrained on the FC3 corpus (indicated with the *-fc* suffix). We provide an overview of all training datasets and hyperparameter configurations used in our experiments in Appendix A.

5 Experiments

5.1 Downstream Performance for Faroese

Experimental Setup. In addition to the models presented in §4, we make use of the monolingual Icelandic model IceBERT and the massively multilingual XLM-on-RoBERTa (XLM-R).[3] We evaluate the performance of this set of pretrained models in several downstream tasks in Faroese: Part-of-Speech tagging (POS), Dependency Parsing (DP) (UD datasets introduced in §2), Named Entity Recognition (NER), and Semantic Text Similarity (i.e., the new NER and STS datasets introduced in §3). For all downstream tasks the task-specific training and evaluation data span monolingual Faroese data points only: we carry out the experimentation via ten-fold cross-validation on the respective Faroese datasets.[4] For each model and downstream task, we carry out ten runs with different random seeds (each run trains the model for 5 epochs with batches of 16 instances) and report the average performance across runs. The exception is the STS training in which the models were fine-tuned for 3 epochs (with training batches of size 8).[5]

Results and Discussion. Table 1 summarizes the results across the four downstream tasks. The best-performing model for POS, as evaluated on the Sosialurin POS corpus, is ScandiBERT-fc3, outperforming ScandiBERT by more than 1 point in terms of F1. However, the ScandiBERT-no-fo-fc3 model, without any Faroese data at pretraining, obtains fully on-par performance with the variant that does include Faroese data.

The best-performing model for NER, and STS is the ScandiBERT-no-fo-fc3 model. Somewhat surprisingly, we get the best performance for the model that does not include any Faroese data in the initial pretraining, that is, it does not adjust the tokenizer/vocabulary to Faroese. Put simply, we observe slight gains over the ScandiBERT-fc3 model. We hypothesize that this might be due to the fact that including Faroese in the vocabulary results in a lower subword overlap with the other Scandinavian languages, which in consequence, slightly reduces the potential for transfer. While there is only a difference of 95 tokens between the two vocabularies, the difference yields 6% of the words in FC3 being tokenized differently.

Finally, the results also demonstrate the importance of focusing on a smaller set of related languages rather than relying on a broader set of languages from the MMTs. Unlike the results from Fujinuma et al. (2022), our results suggest that for languages with higher-resource 'siblings' such as Faroese, a higher-performing LM is a less general ScandiBERT model rather than an MMT such as XLM-R or mBERT. Different variants of ScandiBERT outperform XLM-R without any Faroese data across the board in all evaluation tasks. Another interesting finding is that additionally fine-tuning on Faroese data (the `-fc3` variants) has a much stronger positive impact on XLM-R as the underlying model than on ScandiBERT. Put simply, the importance of in-target language data decreases with the availability of more focused pretained LMs covering only languages related to the target language.

5.2 Additional Experiments

Transfer with Wechsel. To put our work in further context, beyond comparison to MMTs, we consider an alternative transfer learning approach, the Wechsel method (Minixhofer et al., 2022), a recent well-performing method for transferring monolingual Transformers to a new language. Further details and results are presented in Appendix B: they all show far worse performance than those presented

[3] We use the base-sized XLM-R: `https://huggingface.co/xlm-roberta-base`.

[4] Note that our study aims to establish how different pretraining strategies – and in particular languages included in pretraining – affect the models' downstream Faroese performance, rather than to investigate the downstream cross-lingual transfer. One could, naturally, additionally incorporate task-specific data in other Scandinavian languages (and also in English and other languages) in downstream training (i.e., perform cross-lingual transfer for the downstream task).

[5] Due to the limited size of the Faroese dataset, longer training with larger batch size consistently led to overfitting.

Model	POS		NER		UD FP		UD oft		STS
	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	Acc.
IceBERT	85.5 ± 0.19	85.2 ± 0.16	87.9 ± 0.54	96.4 ± 0.09	93.6 ± 0.06	94.6 ± 0.03	92.7 ± 0.32	94.2 ± 0.25	70.6 ± 1.9
IceBERT-fc3	90.9 ± 0.06	90.4 ± 0.06	90.9 ± 0.41	98.9 ± 0.03	96.6 ± 0.06	97.1 ± 0.06	95.3 ± 0.38	96.1 ± 0.32	72.9 ± 1.8
DanskBERT	73.4 ± 0.19	74.3 ± 0.16	85.6 ± 0.44	98.4 ± 0.06	86.2 ± 0.16	87.7 ± 0.09	84.8 ± 0.57	88.7 ± 0.44	73.2 ± 1.3
DanskBERT-fc3	87.1 ± 0.13	86.4 ± 0.13	89.7 ± 0.54	98.8 ± 0.06	96.0 ± 0.06	96.6 ± 0.03	94.2 ± 0.28	95.7 ± 0.19	75.3 ± 1.1
XLM-R	84.6 ± 0.28	85.0 ± 0.28	87.8 ± 0.47	96.3 ± 0.06	93.5 ± 0.06	94.3 ± 0.03	91.5 ± 0.44	93.6 ± 0.35	69.5 ± 2.1
XLM-R-fc3	91.2 ± 0.09	91.2 ± 0.09	90.9 ± 0.41	**98.9 ± 0.06**	97.3 ± 0.06	97.7 ± 0.03	95.7 ± 0.22	96.8 ± 0.19	69.2 ± 2.1
ScandiBERT-no-fo	88.4 ± 0.09	88.1 ± 0.09	89.9 ± 0.25	96.7 ± 0.16	95.9 ± 0.06	96.4 ± 0.06	93.8 ± 0.35	95.0 ± 0.32	75.3 ± 1.5
ScandiBERT-no-fo-fc3	91.5 ± 0.09	91.2 ± 0.09	**91.4 ± 0.35**	98.8 ± 0.06	**97.4 ± 0.03**	**97.8 ± 0.03**	**96.3 ± 0.22**	**96.8 ± 0.19**	**76.5 ± 1.3**
ScandiBERT	90.3 ± 0.09	90.0 ± 0.13	90.2 ± 0.28	99.0 ± 0.06	96.5 ± 0.06	97.1 ± 0.03	95.2 ± 0.32	96.2 ± 0.25	46.3 ± 6.3
ScandiBERT-fc3	**91.6 ± 0.06**	**91.3 ± 0.09**	91.0 ± 0.35	99.0 ± 0.03	97.3 ± 0.06	97.7 ± 0.06	95.9 ± 0.25	96.7 ± 0.22	63.8 ± 6.2

Table 1: Results for all downstream tasks in Faroese using the different base language models, with and without continued Faroese pre-training. The -fc3 postfix indicates models that were further pretrained on FC3. Standard error intervals are also reported.

in Table 1. We hypothesize this is due to how closely related the languages we consider are, as opposed to the distant languages considered in the original Wechsel work.

Task-Specific Transfer. To explore the potential for task-specific transfer between closely related languages, we consider if labelled Scandinavian datasets can be combined to benefit Faroese. In particular, we look at NER as there is an easy way to map between labels of the different languages. See Appendix C for more details. The best result is achieved when training directly from the IceBERT model, which has been trained on the large MIM-GOLD-NER dataset, showing that given enough resources and a close enough language model, such a direct approach can be the most effective.

Further Discussion. Some of the results in Table 1 are as expected. Starting from the closest language relative, the Icelandic model, IceBERT, results in better performance for all downstream tasks than starting with the Danish model DanskBERT. The ScandiBERT model performs better than the massively multilingual XLM-R on all tasks, bar the more semantic FO-STS task.

What is more interesting is that the ScandiBERT-no-fo model that is not trained on Faroese outperforms the model that has Faroese included, when fine-tuned further on the FC3 dataset. In particular, for the higher level Fo-STS task. We hypothesize that this forces the Faroese adaptation to use the word segmentations from the related languages for a higher transfer benefit, as the tokenizing vocabulary was trained without Faroese. This is something we hope to investigate more in future work.

6 Conclusion and Future Work

We have shown that leveraging phylogenetic information and departing from the 'one-size-fits-all' paradigm can improve cross-lingual transfer to low-resource languages. Our evaluation results show that we can substantially improve the transfer performance to Faroese by exploiting data and models of closely-related high-resource languages instead of relying on MMTs. In future work, we hope to extend the investigations and methodology beyond Faroese, to other low-resource languages for which higher-resource language relatives exist.

In order to boost and guide future research on Scandinavian languages in general and Faroese in particular, we make the models *ScandiBERT*[6], *ScandiBERT-no-fo*[7], *DanskBERT*[8] and *FoBERT (ScandiBERT-no-fo-fc3)*[9] available. As well as the new datasets *FC3*[10], *FoNE*[11], and *Fo-STS*[12].

Acknowledgments

VS is supported by the Pioneer Centre for AI, DNRF grant number P1. The work of IV has been supported by a personal Royal Society University Research Fellowship (no 221137; 2022-).

We would like to thank Haukur Barri Símonarson for his comments on the work in its early stages. We also thank Prof. Dr.-Ing. Morris Riedel and his team for providing access to the DEEP cluster at Forschungszentrum Jülich.

References

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages.

[6] https://huggingface.co/vesteinn/ScandiBERT

[7] https://huggingface.co/vesteinn/ScandiBERT-no-faroese

[8] https://huggingface.co/vesteinn/DanskBERT

[9] https://huggingface.co/vesteinn/FoBERT

[10] https://huggingface.co/datasets/vesteinn/FC3

[11] https://huggingface.co/datasets/vesteinn/sosialurin-faroese-ner

[12] https://huggingface.co/datasets/vesteinn/faroese-sts

In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Lars Ahrenberg, Johan Frid, and Leif-Jöran Olsson. 2020. A new resource for swedish named-entity recognition.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

James Barry, Joachim Wagner, and Jennifer Foster. 2019. Cross-lingual parsing with polyglot training and multi-treebank learning: A Faroese case study. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 163–174, Hong Kong, China. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Stian Rødven Eide, Nina Tahmasebi, and Lars Borin. 2016. The swedish culturomics gigaword corpus: A one billion word swedish reference dataset for nlp.

Fahim Faisal and Antonios Anastasopoulos. 2022. Phylogeny-inspired adaptation of multilingual models to new languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 434–452, Online only. Association for Computational Linguistics.

Mikel L. Forcada and Francis M. Tyers. 2016. Apertium: a free/open source platform for machine translation and basic language technology. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation: Projects/Products*, Riga, Latvia. Baltic Journal of Modern Computing.

Yoshinari Fujinuma, Jordan Boyd-Graber, and Katharina Kann. 2022. Match the script, adapt if multilingual: Analyzing the effect of multilingual pretraining on cross-lingual transferability. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1500–1512, Dublin, Ireland. Association for Computational Linguistics.

Evangelia Gogoulou, Ariel Ekgren, Tim Isbister, and Magnus Sahlgren. 2022. Cross-lingual transfer of monolingual models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 948–955, Marseille, France. European Language Resources Association.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

René Haas and Leon Derczynski. 2021. Discriminating between similar nordic languages. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 67–75, Kiyv, Ukraine. Association for Computational Linguistics.

Hinrik Hafsteinsson and Anton Karl Ingason. 2021. Towards cross-lingual application of language-specific PoS tagging schemes. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 321–325, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Zakaris Svabo Hansen, Heini Justinussen, and Mortan Ólason. 2004. Marking av teldutøkum tekstsavni [tagging of a digital text corpus].

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Rasmus Hvingelby, Amalie Brogaard Pauli, Maria Barrett, Christina Rosted, Lasse Malm Lidegaard, and Anders Søgaard. 2020. DaNE: A named entity resource for Danish. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4597–4604, Marseille, France. European Language Resources Association.

Anton Karl Ingason, Eiríkur Rögnvaldsson, Einar Freyr Sigurðsson, and Joel C. Wallenberg. 2012. Faroese parsed historical corpus (FarPaHC) 0.1. CLARIN-IS.

Svanhvít L. Ingólfsdóttir, Ásmundur A. Guðjónsson, and Hrafn Loftsson. 2020. Named entity recognition for icelandic: Annotated corpus and models. In *Statistical Language and Speech Processing*, pages 46–57, Cham. Springer International Publishing.

Fredrik Jørgensen, Tobias Aasmoe, Anne-Stine Ruud Husevåg, Lilja Øvrelid, and Erik Velldal. 2020. NorNE: Annotating named entities for Norwegian. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4547–4556, Marseille, France. European Language Resources Association.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Per Kummervold, Freddy Wetjen, and Javier de la Rosa. 2022. The Norwegian colossal corpus: A text corpus for training large Norwegian language models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3852–3860, Marseille, France. European Language Resources Association.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Barun Patra, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R. Gormley, and Graham Neubig. 2019. Bilingual lexicon induction with semi-supervision in non-isometric embedding spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 184–193, Florence, Italy. Association for Computational Linguistics.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Edoardo Maria Ponti, Julia Kreutzer, Ivan Vulic, and Siva Reddy. 2021. Modelling latent translations for cross-lingual transfer. *CoRR*, abs/2107.11353.

Fabian David Schmidt, Ivan Vulić, and Goran Glavaš. 2022. Don't stop fine-tuning: On training regimes for few-shot cross-lingual transfer with multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10725–10742, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Vésteinn Snæbjarnarson, Haukur Barri Símonarson, Pétur Orri Ragnarsson, Svanhvít Lilja Ingólfsdóttir, Haukur Jónsson, Vilhjalmur Thorsteinsson, and Hafsteinn Einarsson. 2022. A warm start and a clean crawled corpus - a recipe for good language models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4356–4366, Marseille, France. European Language Resources Association.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788, Melbourne, Australia. Association for Computational Linguistics.

Leon Strømberg-Derczynski, Manuel Ciosici, Rebekah Baglini, Morten H. Christiansen, Jacob Aarup Dalsgaard, Riccardo Fusaroli, Peter Juel Henrichsen, Rasmus Hvingelby, Andreas Kirkedal, Alex Speed Kjeldsen, Claus Ladefoged, Finn Årup Nielsen, Jens Madsen, Malte Lau Petersen, Jonathan Hvithamar Rystrøm, and Daniel Varab. 2021. The Danish Gigaword corpus. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 413–421, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Francis Tyers, Mariya Sheyanova, Aleksandra Martynova, Pavel Stepachev, and Konstantin Vinogorodskiy. 2018. Multi-source synthetic treebank creation for improved cross-lingual dependency parsing. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 144–150, Brussels, Belgium. Association for Computational Linguistics.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, Gertjan van Noord, and Sebastian Ruder. 2022. Hyper-X: A unified hypernetwork for multi-task multilingual transfer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7934–7949, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 863–877, Dublin, Ireland. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Haoran Xu and Kenton Murray. 2022. Por qué não utiliser alla språk? mixed training with gradient optimization in few-shot cross-lingual transfer. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2043–2059, Seattle, United States. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Mengjie Zhao, Yi Zhu, Ehsan Shareghi, Ivan Vulić, Roi Reichart, Anna Korhonen, and Hinrich Schütze. 2021. A closer look at few-shot crosslingual transfer: The choice of shots matters. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5751–5767, Online. Association for Computational Linguistics.

A Training of language models

We train new BPE vocabularies for all the new models we train, ScandiBERT, ScandiBERT-no-fo, and

DanskBERT. All models use the same vocabulary size of 50k. The ScandiBERT vocabulary is trained using all the languages, the ScandiBERT-no-fo vocabulary is trained without the Faroese data, and the DanskBERT vocabulary is only trained on the Danish text. Vocabularies are trained using the SentencePiece software (Kudo and Richardson, 2018), and character coverage is set to 99.995 %.

Pre-training of the new language models is done using `fairseq` (Ott et al., 2019) using the RoBERTa-base (Liu et al., 2019) configuration, fine-tuning is done using the `transformers` (Wolf et al., 2020) library. ScandiBERT and ScandiBERT-no-fo were trained for 72 epochs, using a batch size of 8.8k sequences on 24 NVIDIA V100 cards for approximately 14 days each. Initial testing showed that the larger batch size showed better performance than going for around 2k sequences, possibly due to the mixture of differing languages. DanskBERT, on the other hand, similar to IceBERT and RoBERTa showed better performance at the smaller batch size. DanskBERT was trained to convergence for 500k steps using 16 V100 cards for approximately 14 days.

All *-fc* models are further trained for 50 epochs, with an effective batch size of 100k tokens for 12k updates, over the FC3 dataset for Faroese adaptation.

An overview of the data used to train the language models is shown in Table 2. For details on the Icelandic data, we refer to (Snæbjarnarson et al., 2022). For the other datasets, we refer to §2.

B Wechsel results

We compare our method to another transfer learning approach presented by Minixhofer et al. (2022). The FC3 dataset is used to train fastText embeddings for Faroese, and the Icelandic datasets are used to train fastText embeddings for Icelandic. These embeddings are then used to convert the multilingual models to Faroese using the Wechsel approach. We confirm the quality of the Icelandic embeddings by running an Icelandic semantic evaluation suite adapted from `https://github.com/stofnun-arna-magnussonar/ordgreypingar_embeddings`, showing our embeddings are comparable or of higher quality than those released by Meta (Grave et al., 2018).

Language	Datasets	Size
Icelandic	IGC / IC3 / Skemman / Hirslan	16 GB
Danish	Danish Gigaword Corpus (incl. Twitter)	4,7 GB
Norwegian	NCC corpus	42 GB
Swedish	Swedish Gigaword Corpus	3,4 GB
Faroese	FC3 + Sosialurinn + Bible	69 MB

Table 2: Datasets used to train ScandiBERT, ScandiBERT-no-fo and DanskBERT

The experiments in Table 3 all show sub-par performance compared to the results in non-Wechsel results in Table 1. The Wechsel work considers transfer from English-dominant models, GPT2 and RoBERTa to French, German, Chinese, Swahili, Sundanese, Scottish Gaelic, Uyghur and Malagasy. None of which are closely related to English. One reason for the discrepancy in the results could be that the shuffling of the embedding matrix to convert it is more catastrophic when considering close languages. Another reason could be that both Faroese and Icelandic are morphologically rich and that all variants of the words were not properly mapped during the conversion of the embedding matrix.

C Mapping NER datasets

The datasets used to create a Scandinavian NER-corpus are DaNE (Danish), FoNE (Faroese), MIM-GOLD-NER (Icelandic), NorNE (Norwegian), and SWE-Nerc (Swedish), presented in §2. The results in Table 4 show that the best result is obtained when training directly from the IceBERT model. The ScandiBERT model has a higher variance when pre-fine-tuned on the combined NER corpora. This approach could also be made directly for the UD corpus, POS (in particular, using the Icelandic POS data), and other corpora as they become available for training or evaluation in Faroese. This demonstrates how resources from a related language can substantially benefit a low-resource language.

To combine the NER datasets, we map the tags to the CoNLL schema used by the Icelandic MIM-GOLD-NER and the Faroese FoNE datasets. The Danish DaNE dataset uses a subset of the tags used for Icelandic and Faroese, so the mapping is purely nominal. The mapping for Norwegian (NorNE) and Swedish (SweNERC) datasets is shown in Table 5.

D Prior Work on Transfer learning for Faroese

We know of three works that consider transfer learning for Faroese from the Scandinavian languages.

	Model	POS F1	POS Acc.	NER F1	NER Acc.	UD FP F1	UD FP Acc.	UD oft F1	UD oft Acc.	STS Acc.
Wechsel	IceBERT	74.4 ± 0.16	75.7 ± 0.16	67.7 ± 1.2	98.7 ± 0.06	83.6 ± 0.35	84.6 ± 0.38	66.6 ± 9.01	75.7 ± 5.88	27.3 ± 3.9
	IceBERT-fc3	89.0 ± 0.06	89.4 ± 0.09	88.5 ± 0.47	96.4 ± 0.09	96.3 ± 0.03	96.5 ± 0.03	95.6 ± 0.28	96.2 ± 0.25	67.7 ± 2.8
	XLM-R	68.9 ± 0.16	73.5 ± 0.13	59.7 ± 0.92	99.0 ± 0.06	81.0 ± 0.19	84.5 ± 0.13	71.8 ± 0.73	79.7 ± 0.44	11.4 ± 4.6
	XLM-R-fc3	86.8 ± 0.09	88.7 ± 0.09	88.8 ± 0.41	98.4 ± 0.06	96.3 ± 0.03	96.7 ± 0.03	95.6 ± 0.25	96.5 ± 0.19	65.7 ± 2.8
	ScandiBERT-no-fo	71.3 ± 0.16	72.5 ± 0.16	65.1 ± 0.54	98.8 ± 0.03	82.0 ± 0.19	83.4 ± 0.19	75.0 ± 0.66	81.0 ± 0.57	29.4 ± 4.8
	ScandiBERT-n.f.-fc3	89.2 ± 0.06	89.6 ± 0.06	89.2 ± 0.54	99.0 ± 0.03	96.8 ± 0.06	97.1 ± 0.06	96.1 ± 0.28	96.8 ± 0.22	74.7 ± 1.0
	ScandiBERT	72.6 ± 0.28	73.8 ± 0.28	65.7 ± 0.54	98.8 ± 0.03	83.0 ± 0.38	84.0 ± 0.28	76.8 ± 0.51	82.5 ± 0.35	8.7 ± 5.3
	ScandiBERT-fc3	89.3 ± 0.09	89.7 ± 0.09	88.8 ± 0.54	98.7 ± 0.06	96.8 ± 0.03	97.1 ± 0.03	96.0 ± 0.25	96.7 ± 0.25	53.6 ± 6.0

Table 3: Results for all downstream tasks using different base language models after Wechsel adaptation, with and without continued Faroese pre-training. The results are significantly worse than without Wechsel adaptations.

Model	Pre-ft.	Ft.	F1	Acc.
SB-no-fo-fc3	None	Yes	91.4 ± 0.35	98.8 ± 0.06
ScandiBERT	Icel.	Yes	**92.0 ± 0.32**	98.8 ± 0.06
ScandiBERT	All	No	91.5 ± 0.51	98.9 ± 0.06
ScandiBERT	All	Yes	91.8 ± 0.51	99.0 ± 0.06
XLM-R	All	No	90.6 ± 0.19	99.0 ± 0.03
XLM-R	All	Yes	90.8 ± 0.47	99.0 ± 0.06

Table 4: NER performance when models are pre-finetuned on all Scandinavian datasets and then fine-tuned on FoNER.

Language	Original	Mapped
Norwegian	O	O
Norwegian	PER	Person
Norwegian	ORG	Organization
Norwegian	GPE_LOC	Location
Norwegian	PROD	Miscellaneous
Norwegian	LOC	Location
Norwegian	GPE_ORG	Organization
Norwegian	DRV	O
Norwegian	EVT	Miscellaneous
Norwegian	MISC	Miscellaneous
Swedish	O	O
Swedish	EVN	Miscellaneous
Swedish	GRO	Organization
Swedish	LOC	Location
Swedish	MNT	Miscellaneous
Swedish	PRS	Person
Swedish	TME	Time
Swedish	WRK	Miscellaneous
Swedish	SMP	Miscellaneous

Table 5: Mapping of tags to create a unified NER dataset for the Scandinavian languages.

In (Tyers et al., 2018), a rule-based translation system (Apertium (Forcada and Tyers, 2016)) is used to translate the Faroese Wikipedia into Swedish, Norwegian Bokmål, and Norwegian Nynorsk. The translations are then aligned, and the translations dependency-parsed. The resulting trees are then mapped to the original Faroese sentences and used for POS-tagging and annotating morphological features. The second work is a mapping between Faroese and Icelandic POS-tags (Hafsteinsson and Ingason, 2021); while not a direct application, the authors suggest the mapping may be of use for transfer learning between the languages. Finally, (Barry et al., 2019) use machine translation and dependency parsing for cross-lingual syntactic knowledge transfer from Danish, Norwegian, and Swedish to Faroese.

Evaluating Morphological Generalisation in Machine Translation by Distribution-Based Compositionality Assessment

Anssi Moisio
Department of Information and Communications Engineering
Aalto University, Finland
anssi.moisio@aalto.fi

Mathias Creutz
Department of Digital Humanities
University of Helsinki, Finland
mathias.creutz@helsinki.fi

Mikko Kurimo
Department of Information and Communications Engineering
Aalto University, Finland
mikko.kurimo@aalto.fi

Abstract

Compositional generalisation refers to the ability to understand and generate a potentially infinite number of novel meanings using a finite group of known primitives and a set of rules to combine them. The degree to which artificial neural networks can learn this ability is an open question. Recently, some evaluation methods and benchmarks have been proposed to test compositional generalisation, but not many have focused on the morphological level of language. We propose an application of the previously developed *distribution-based compositionality assessment* method to assess morphological generalisation in NLP tasks, such as machine translation or paraphrase detection. We demonstrate the use of our method by comparing translation systems with different BPE vocabulary sizes. The evaluation method we propose suggests that small vocabularies help with morphological generalisation in NMT.[1]

1 Introduction

Natural languages usually adhere to the *principle of compositionality*, with the exception of idiomatic expressions. Partee et al. (1995) phrased this principle as "The meaning of a whole is a function of the meanings of the parts and of the way they are syntactically combined". Deriving from this principle, *compositional generalisation* (CG) refers to the capacity to understand and generate a potentially infinite number of novel meanings using a finite group of known primitives and a set of rules of how to combine them. In the case of language, morphemes are combined into words and words in turn into phrases and sentences, using the syntactical rules of the language.

Neural networks have long been argued to lack the ability to generalise compositionally the way humans do (Fodor and Pylyshyn, 1988; Marcus, 1998). After the rapid improvement of neural NLP systems during the previous decade, this question has gained renewed interest. Many new evaluation methods have been developed to assess whether the modern sequence-to-sequence (seq2seq) architectures such as Transformers exhibit CG, since they certainly exhibit increasingly competent linguistic behaviour. For instance, in one of the seminal CG evaluation methods, called SCAN (Lake and Baroni, 2018), a seq2seq system has seen certain natural language commands in training and needs to combine them in novel ways in testing.

CG is a general capacity that can be seen as a desideratum in many NLP tasks, and in machine learning more generally. Furthermore, CG is a multifaceted concept that can be, and should be, decomposed into narrower, more manageable aspects that can be tested separately (Hupkes et al., 2020). For example, NLP systems should be able to generalise compositionally both on the level of words and on the level of morphology.

Although many aspects of CG have recently been evaluated in NLP (an extensive review is offered by Hupkes et al. (2022)), some aspects have remained without an evaluation method. We identify (see Section 2) a lack of methods to

[1]Code and datasets available at https://github.com/anmoisio/morphogen-dbca

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 738–751

evaluate *compositional morphological* generalisation using only *natural*, non-synthetic, data. To fill this gap, we propose an application of the *distribution-based compositionality assessment* (DBCA) method (Keysers et al., 2020) (henceforth *Keysers*) to generate adversarial data splits to evaluate morphological generalisation in NLP systems.

Specifically, we split natural language corpora while controlling the distributions of lemmas and morphological features (*atoms* in the terminology of *Keysers*) on the one hand, and the distributions of the combinations of atoms (*compounds*, not to be confused with compound words) on the other hand. By requiring a low divergence between the atom distributions of the train and test sets, and a high divergence between the compound distributions, we can evaluate how well a system is able to generalise its morphological knowledge to unseen word forms.

For example, if our corpus included as atoms the lemmas "cat" and "dog", and the morphological tags `Number=Sing` and `Number=Plur`, a low divergence between the atom distributions would mean that both the training and test sets included all four of the atoms, and a high compound divergence would mean that the sets include different combinations of them, for instance training set {`cat, dogs`} and test set {`cats, dog`}.

Our main contributions are the following: **firstly**, we describe an application of DBCA to evaluate morphological generalisation in any NLP task in which the train and test data consist of sentences for which morphological tags are available. **Secondly**, we demonstrate how by this method we can evaluate morphological generalisation in machine translation without manual test design. And **thirdly**, using our proposed method, we assess the effect of the source language BPE (Sennrich et al., 2016) vocabulary size in Finnish-English NMT performance, and conclude that a smaller vocabulary helps the NMT models in morphological generalisation.

2 Background

In the broader field of machine learning, CG has been analysed in various domains besides that of natural language, such as visual question answering (Bahdanau et al., 2018), visual reasoning (Zerroug et al., 2022) and mathematics (Saxton et al., 2019), but in this work we focus on natural language tasks. Two reviews have recently been published about CG in NLP, of which Donatelli and Koller (2023) focus on semantic parsing and the aforementioned Hupkes et al. (2022) (henceforth *Hupkes*) take a broader view, reviewing generalisation in general, not only the compositional type.

Hupkes categorised NLP generalisation experiments along five dimensions, of which we discuss two here to motivate our work. The first is the *type of generalisation* along which the *compositional* type is distinguished from the *morphological* type. *Hupkes* define compositionality as "the ability to systematically recombine previously learned elements to map new inputs made up from these elements to their correct output. In language, the inputs are 'forms' (e.g. phrases, sentences, larger pieces of discourse), and the output that they need to be mapped to is their meaning ...". In NMT, the translation works as a proxy to meaning, so that CG can be evaluated by evaluating the translation (Dankers et al., 2022) (other works that assess CG in NMT include (Li et al., 2021; Raunak et al., 2019)).

Hupkes contrast compositional with structural, including morphological, generalisation where an output space is not required but which focuses on generation of the correct forms. These definitions suggest a clear divide between the categories, which is understandable when analysing the literature: morphological generalisation, specifically inflection generation, has for decades been studied in psycholinguistics (Berko, 1958; Marcus et al., 1992) and computational linguistics (Rumelhart and McClelland, 1986; Corkery et al., 2019; Kodner et al., 2022). These studies do not address the question of how the different inflections are mapped to different meanings, hence they do not address *compositional* generalisation. However, inflections do bear meaning, of course, and so *compositional morphological* generalisation is an ability that humans possess, and NLP systems ought to be tested on.

Although *Hupkes* do not categorise any experiments as assessing *compositional morphological* generalisation, there has been at least one that we think could be so categorised: Burlot and Yvon (2017) designed an NMT test suite in which a single morphological feature is modified in a source language sentence, creating a contrastive pair, and the translations of the contrastive sentences are inspected for a corresponding change in the target

language.

The other dimension of *Hupkes* relevant to the motivation of our experiments is that of *shift source*: the shift between train and test sets could occur naturally (as in two natural corpora in different domains), it can be created by generating synthetic data, or an artificial partition of natural data can be obtained. Most of the previous methods to assess compositional generalisation in NMT (Burlot and Yvon, 2017; Li et al., 2021; Dankers et al., 2022) have synthetised data for the test sets. Generating synthetic data has its benefits: any morphological form can occur in the data when it is generated, and a single morphological feature can be easily focused on and evaluated qualitatively as well as quantitatively.

However, synthetic data has at least practical disadvantages, leaving aside the more theoretical question of how well the synthetic language approximates natural language, assuming the ultimate goal is systems that process natural language. In practice, synthetic test sets require manual design, which means it is difficult to come by a method to generate an unlimited number of synthetic sentences, or a method that could work in arbitrary languages. Furthermore, when manually designing test suites to evaluate morphological generalisation, as Burlot and Yvon (2017) designed, the requirement for manual work restricts the number of morphological phenomena we have resources to test.

The other option is to create artificial data splits of natural data. While natural data may be noisier and it might be more difficult to focus on a specific phenomenon of the language by this method, this method is easier to automate completely. Furthermore, the method of automatically generating data splits that we present in the next section is also generalisable to other tasks (e.g. paraphrase detection) and any corpus of sentences. Generating artificial data splits of natural data has previously been used to test CG in translation (Raunak et al., 2019), as well as to assess the capacity to capture long-distance dependencies in translation (Choshen and Abend, 2019), but not to assess *morphological* generalisation, as far as we are aware. (For a more general discussion of splitting data into non-random testing and training sets, see Søgaard et al. (2021).)

The method we describe in this paper is an application of the DBCA method developed by *Keysers*. Since this method is generic and task-agnostic, it can be applied to any dataset for which it is possible to define atom and compound distributions. Although it is easier to define these distributions for synthetic data, as in the CFQ dataset described by *Keysers*, it can also be applied to natural data, for example in semantic parsing (Shaw et al., 2021). The next section describes how DBCA can be used to assess morphological generalisation in any task where the training and testing corpora consist of natural language sentences.

3 Applying DBCA to assess morphological generalisation in NLP

DBCA is a method to evaluate CG by splitting a dataset into train/test sets with differing distributions, requiring some capacity to generalise from the training distribution to the test distribution. Specifically, the distributions of *atoms* (known primitives) and *compounds* (combinations of atoms) are controlled to get similar atom distributions but contrasting compound distributions in the training and test sets. In our application of DBCA to a corpus of natural language sentences, the atom distribution $\mathcal{F}_A$ of the corpus is the distribution of the lemmas and morphological features and the compound distribution $\mathcal{F}_C$ is the distribution of their combinations. Table 1 presents examples of atoms and compounds in this work.

To determine the atom and compound distributions, we first need to obtain the lemmas and morphological tags of all words in the corpus, which we accomplish for Finnish corpora using the Turku Neural Parser Pipeline (Kanerva et al., 2018). For the experiments presented in Section 4, we use a corpus of 1M sentences. In practice, we do not have resources to control the distribution of all lemmas even in this relatively small corpus, so we need to select some subset of the lemmas that we include in our analysis.

Selecting the lemma subset could be done in many ways, but the following is a way we deemed reasonable. To limit the number of lemmas, we first filter out lemmas that do not appear in the list of 94110 Finnish lemmas[2] or, since this list does not include proper names, in lists[3] of names

[2]Available at `https://kaino.kotus.fi/sanat/nykysuomi/`

[3]List of names of places: `https://kaino.kotus.fi/eksonyymit/?a=aineisto`
English given names: `https://en.wiktionary.org/wiki/Appendix:English_given_names`

	Atoms	Compounds
Desc.	lemmas and morphological tags	combinations of atoms
E.g.	`tunturi`, `Case=Gen`, `Case=Ade`, `Number=Sing`, `Number=Plur`	`tunturi|Case=Gen|Number=Plur` (*tunturien*, of mountains); `tunturi|Case=Ade|Number=Sing` (*tunturilla*, on mountain)

Table 1: Description and examples of what we call "atoms" and "compounds". The compounds are the unique word forms, determined by the lemma and the morphological tags. The word form and its English translation are written inside the brackets.

for places, or lists of Finnish and English given names. This way, the lemmas that are filtered out include most of the typos and other non-words. Then we rank the remaining lemmas by frequency in our corpus, and sample a fixed number of lemma occurrences from constant intervals in the ranked list of lemmas. Specifically, we take 40000 lemma occurrences at intervals of 1000 lemma types in the list of lemmas. For our corpus of 1M sentences, this method subsamples the lemmas with frequency ranks of 1000-1033, 2000-2083, 3000-3174, and so on, so that there are fewer frequent lemma types than rare lemma types, but the total number of occurrences of each bucket is around 40k. Lemmas that occur fewer than 10 times in the corpus are excluded. After the filtering, we have 8720 lemma types that occur about 390k times in total in our corpus of 1M sentences. We append the list of 48 morphological tags[4] (after filtering some that indicate uninteresting words such as 'Typo' and 'Abbr') that these lemmas appear with to the lemma list to complete our list of atoms.

Keysers weighted the compounds to "avoid double-counting compounds that are highly correlated with some of their super-compounds". The idea is to lessen the weight of those compounds that only or often occur as a part of one certain super-compound. We weight the compounds analogously, but use only two levels in our weighting, which makes the weighting simpler than in *Keysers*: we consider the combinations of morphological tags as the lower level of compounds, and these combined with lemmas as the higher level. Thus the motivation for weighting in our case is not to use those morphological tag combinations that only occur with some specific lemma. Therefore, we look for the lemma with which each morphological tag combination occurs most often, and give the tag combination a weight that is the complement of the empirical probability that the tag combination occurs with this lemma. For example, we found that the rare morph tag combination `Case=Ade | Degree=Pos | Number=Plur | PartForm=Pres | VerbForm=Part | Voice=Pass` occurs 84% of the time with the lemma `saada` forming the word "saatavilla", so it gets a weight of 0.16. After weighting the tag combinations, we exclude those that have a weight of 0.33 or less.

After the described filtering steps, we have 8322 atoms, which includes the lemmas and morphological tags. The atoms occur about 1.3M times in 273k sentences in our corpus of 1M sentences. There are 335 morphological tag combinations, which create about 69k unique word forms with the lemmas; i.e. we use 69k compounds in our analysis. These compounds occur 352k times in the corpus.

Calculating atom and compound divergences is done the same way as in *Keysers*. Namely, divergence $\mathcal{D}$ between distributions P and Q is calculated using the Chernoff coefficient $C_\alpha(P\|Q) = \sum_k p_k^\alpha \, q_k^{1-\alpha} \in [0,1]$ (Chung et al., 1989), with $\alpha = 0.5$ for the atom divergence and $\alpha = 0.1$ for the compound divergence. As described by *Keysers*, $\alpha = 0.5$ for the atom divergence "reflects the desire of making the atom distributions in train and test as similar as possible", and $\alpha = 0.1$ for the compound divergence "reflects the intuition that it is more important whether a certain compound occurs in P (train) than whether the probabilities in P (train) and Q (test) match exactly". Since the Chernoff coefficient is a similarity metric, the atom and compound divergences of a train set V and a test set W are:

$$\mathcal{D}_A(V\|W) = 1 - C_{0.5}(\mathcal{F}_A(V) \,\|\, \mathcal{F}_A(W))$$
$$\mathcal{D}_C(V\|W) = 1 - C_{0.1}(\mathcal{F}_C(V) \,\|\, \mathcal{F}_C(W)).$$

and Finnish: `https://tinyurl.com/3mn52ms6` `https://tinyurl.com/mwjvaxkk`

[4]See `https://universaldependencies.org/docs/fi/feat/` for the list of Finnish morphological tags.

Procedure 1 Data division algorithm.

Input: G ▷ Corpus of sentences
Input: N ▷ Use N sentences from G
Input: a ▷ Lower bound for $|V|/|W|$
Input: b ▷ Upper bound for $|V|/|W|$
Output: V, W ▷ Train set, test set
  $V \leftarrow \{x \in_R G\}$ ▷ A random sentence
  $W \leftarrow \emptyset$
  $G \leftarrow G \backslash V$
  for $i \leftarrow 1$ **to** N **do**
    $r \leftarrow |V|/|W|$
    $s_V \leftarrow \max_{x \in G} \text{score}(V \cup \{x\}, W)$
    $i_V \leftarrow \text{argmax}_{x \in G} \text{score}(V \cup \{x\}, W)$
    $s_W \leftarrow \max_{x \in G} \text{score}(V, W \cup \{x\})$
    $i_W \leftarrow \text{argmax}_{x \in G} \text{score}(V, W \cup \{x\})$
    if $(s_V > s_W \wedge r < b) \vee r < a$ **then**
      $V \leftarrow V \cup \{i_V\}$
      $G \leftarrow G \backslash \{i_V\}$
    else
      $W \leftarrow W \cup \{i_W\}$
      $G \leftarrow G \backslash \{i_W\}$
    end if
  end for

Once the divergences are defined, we can split a corpus of natural language sentences into training and testing sets with an arbitrary compound and atom divergence values. For this, we use a simple greedy algorithm, sketched in Algorithm 1. For a maximum compound divergence split, the score is calculated as

$$\text{score}(Q, P) = \mathcal{D}_C(Q\|P) - \mathcal{D}_A(Q\|P),$$

and in general, for any desired compound divergence value c:

$$\text{score}(Q, P) = -|c - \mathcal{D}_C(Q\|P)| - \mathcal{D}_A(Q\|P).$$

In practice, we do not have resources to calculate the $\max_{x \in G}$ score. Instead, at each iteration we take a subset $G' \subset G$, say 1000 sentences, and calculate $\max_{x \in G'}$ score.

As mentioned above, this method can be used for any corpus that consists of natural language sentences for which the morphological tags can be obtained. In the next section we use this method to assess morphological generalisation in machine translation.

4 Experiments and results

4.1 NMT model training setup and data

We chose Finnish as the language we analyse because of its rich morphology and because there is a good morphological tagger available for Finnish. We use the English-Finnish parallel corpus from the Tatoeba challenge data release (Tiedemann, 2020). We first apply some heuristics provided by Aulamo et al. (2020) to remove noisy data, and restrict the maximum sentence length to 100 words, after which we take a random sample of 1 million sentence pairs.

We use the OpenNMT-py (Klein et al., 2017) library to train Finnish-English Transformer NMT models using the hyperparameters provided in the example config file[5], which includes the standard 6 transformer layers with 8 heads and a hidden dimension of 512, as in (Vaswani et al., 2017). We train the models until convergence or until a maximum of 33000 steps with 2000 warm-up steps and a batch size of 4096 tokens.

For more details about the setup, see the Github repository linked on the first page.

4.2 The effect of compound divergence on translation performance

The basic experiment we propose is to make at least two different train/test splits of a corpus, using $\mathcal{D}_C$ values of 0 and 1, respectively, (keeping $\mathcal{D}_A = 0$) and assess the change in translation performance (for which we use BLEU (Papineni et al., 2002) and chrF2++ (Popović, 2017) as metrics). Since with $\mathcal{D}_C = 1$ there are more unseen word forms in the test set, we expect a decrease in translation performance from $\mathcal{D}_C = 0$ to $\mathcal{D}_C = 1$ that is caused by the $\mathcal{D}_C = 1$ test set requiring more morphological generalisation capacity.

We show empirically the decrease in performance in Section 4.3, but the cause of this decrease is of course more difficult to verify exactly. The atom and compound distributions are the only things we explicitly control when splitting the corpus, and we only require the compound divergence to differ between different data splits. Therefore, we assume the differing compound divergence to be the cause of this effect, but to be more certain, we conduct two simple checks to look for confounding factors.

[5] https://github.com/OpenNMT/OpenNMT-py/blob/9d617b8b/config/config-transformer-base-1GPU.yml

Firstly, an increase in the average sentence length could be another factor that makes one test set more difficult than another. Increasing the sequence length from training to test set is actually a method that has been proposed to test a certain type of compositional generalisation, sometimes called *productivity* (Hupkes et al., 2020; Raunak et al., 2019). We calculated the average sentence lengths of the train and test sets of the 8 different data splits that we obtained using 8 different random seeds for the data split algorithm. What we found is that for $\mathcal{D}_C = 1$ the average lengths in test sets are actually shorter (ranging from 11.35 to 11.66 words) than those for $\mathcal{D}_C = 0$ (ranging from 12.27 to 13.72 words). The average training set sentence lengths are similar for both $\mathcal{D}_C$ values, ranging from 8.66 to 8.79 for $\mathcal{D}_C = 0$ and from 8.65 to 8.73 for $\mathcal{D}_C = 1$. Thus we know that an increased difference between train and test set sentence lengths cannot explain the decrease in NMT performance from $\mathcal{D}_C = 0$ to $\mathcal{D}_C = 1$ since the difference is actually larger for $\mathcal{D}_C = 0$. The fact that the average sentence length in training sets is always significantly shorter than in test sets is an interesting unintended artefact of the data division algorithm that deserves further investigation in the future, but it does not confound our analysis.

As the second sanity check, we evaluated the NMT models on a neutral test set to see if, for any reason, the training set would be in general worse with $\mathcal{D}_C = 1$ than with $\mathcal{D}_C = 0$, instead of only being worse for the specific test set that we have created. For this we used the Tatoeba challenge test set, which we did not use to train or tune the hyperparameters of any models. The results for the vocabulary size 1000 are presented in Figure 1. We used the models trained on the training sets from the data splits with compound divergences 0.0, 0.5 and 1.0. The compound divergences between these training sets and the Tatoeba challenge test set do correlate with the target $\mathcal{D}_C$ of the data split, but they range only from about 0.4 to 0.6.

From Figure 1 we can see that the NMT models trained with different data sets, from data splits with different $\mathcal{D}_C$ values, do not show similar decrease in performance on the neutral-ish Tatoeba challenge test set as on the test sets obtained from the data split algorithm. We take this to mean that the models trained on $\mathcal{D}_C = 1$ data splits are not in general worse than those trained with $\mathcal{D}_C = 0$ data splits, but only worse on the high-divergence test set.

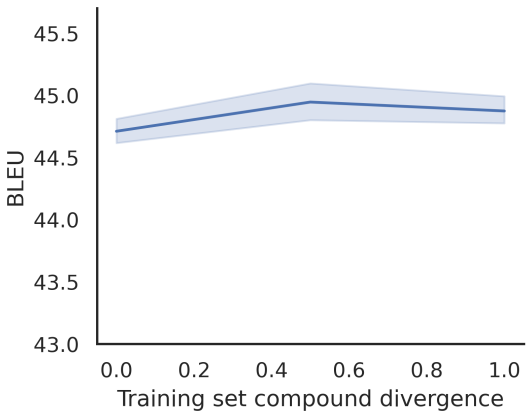


Figure 1: Results on the Tatoeba challenge test set. The x-axis labels denote the compound divergences between the training sets and the test sets analysed later in Figure 2. That is, the divergence is not between the training sets and the Tatoeba challenge test set.

4.3 The effect of BPE vocabulary size on morphological generalisation in NMT

Next, we make the assumption, based on the analysis in Section 4.2, that we can measure morphological generalisation by measuring the decrease of NMT performance between train/test splits of $\mathcal{D}_C = 0$ and $\mathcal{D}_C = 1$. Previous studies have suggested the hypothesis that NMT models with smaller BPE vocabularies are more capable of modelling morphological phenomena than those with larger vocabularies (for example Libovický and Fraser (2020)). In this section, we compare the morphological generalisation capacities of NMT models with different source-side (Finnish) vocabulary sizes, using the method we have proposed.

As a preliminary experiment, we tuned the BPE vocabulary size for our setup (see Section 4.1) on the Tatoeba challenge development set, and found the optimal size to be around 3000 BPE tokens for both the source and target languages. Since we are interested in the Finnish morphology, next we kept the target (English) vocabulary size constant and varied only the source-side vocabulary size.

One thing to note about the vocabulary size is that when we train an NMT system keeping the number of tokens in each batch constant, the number of steps until convergence usually decreases

when the vocabulary size increases, since one epoch takes fewer steps. This reduction in compute, when using a larger vocabulary, is to some extent compensated by the increase of the input layer size (and output layer size, if target language vocabulary is increased too).

We chose 7 different vocabulary sizes, 3 larger and 3 smaller than the optimal 3000, and evaluated them with target compound divergence values of 0.0, 0.25, 0.5, 0.75 and 1.0. The sizes of the test sets are in the order of a few tens of thousands, or a little over a hundred thousand, sentences. The relatively large test set size leads to statistical significance even for small BLEU differences (see Table 3 for details).

From the BLEU results for $\mathcal{D}_C = 0$ and $\mathcal{D}_C = 1$ in Figure 2 we can see that the BLEU results drop, as expected, when the test set demands (more) capacity to generalise to unseen morphological forms. Furthermore, when comparing the different vocabulary sizes, we can notice that as we either increase or decrease the vocab size from 3000, the performance drops, but it drops slightly differently w.r.t $\mathcal{D}_C$. This effect is most conspicuous for the pair of sizes 500 and 18000. The larger vocabulary performs slightly better when there is less need for morphological generalisation, but the small vocabulary performs better when it is needed more. In general, from this figure we can see that the vocabulary size roughly correlates with the angle of the downward slope, suggesting that the larger the vocabulary, the poorer the capacity for morphological generalisation.

To investigate the effect of the initialisation of the data split algorithm on the results, we split the same corpus starting from 8 different random initialisations, and trained NMT models for each data split. For this, we chose two pairs of vocabulary sizes that showed most clearly contrasting performance w.r.t $\mathcal{D}_C$: 500&18000 and 1000&6000. The main results are presented in Table 2. For these results, the test sets of the 8 random seeds are concatenated together to create exceptionally large test sets of around 400k-500k sentences. The results for the individual data splits are presented in Appendix A in Table 4.

From these results we can see the same contrasting performance of the small and large vocabularies w.r.t the different compound divergence values. The difference is small but statistically significant. The models with small vocabularies show better performance than those with large ones when morphological generalisation is needed, and vice versa when morphological generalisation is not needed as much.

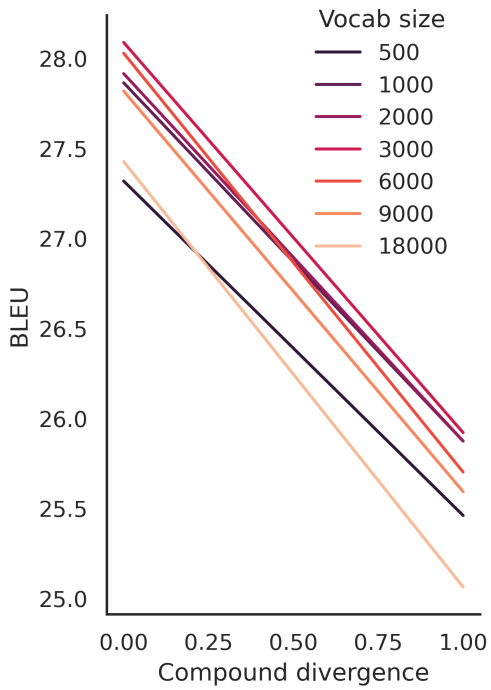


Figure 2: Different source vocabulary sizes evaluated with minimum and maximum (0 and 1) compound divergence data splits. Compound divergence value 1 requires more morphological generalisation. The larger the vocabulary the steeper the slope, suggesting poorer ability to generalise. For more details, see Table 3 in Appendix A.

5 Discussion and future work

In Section 3, we proposed an application of DBCA to divide any corpus of sentences, for which morphological tags are available, into training and test sets with similar distributions of lemmas and morphological tags but contrasting distributions of word forms, in order to assess morphological generalisation. By this method, we can take a large proportion of the morphological phenomena of a selected language into consideration, in our experiments 335 different morphological categories that together with about 8k lemmas create 69k unique Finnish word forms, and evaluate the effects of the contrasting train/test distributions of the word forms in machine translation. This enables a different, complementing type of assessment of morphological generalisation than previous synthetic

Vocab	chrF2++ $\mathcal{D}_C = 0$	chrF2++ $\mathcal{D}_C = 1$	BLEU $\mathcal{D}_C = 0$	BLEU $\mathcal{D}_C = 1$
500	51.20 (51.20 ± 0.05)	**49.33** (49.33 ± 0.05)	27.50 (27.50 ± 0.07)	**25.4** (25.40 ± 0.07)
18000	**51.29** (51.29 ± 0.05)	49.04 (49.05 ± 0.05)	**27.69** (27.69 ± 0.07)	25.18 (25.18 ± 0.07)
	$p = 0.0003$	$p = 0.0003$	$p = 0.0003$	$p = 0.0003$
1000	51.78 (51.78 ± 0.05)	**49.79** (49.79 ± 0.05)	28.17 (28.17 ± 0.07)	**25.89** (25.89 ± 0.07)
6000	**51.83** (51.83 ± 0.05)	49.67 (49.67 ± 0.05)	**28.24** (28.24 ± 0.07)	25.80 (25.80 ± 0.07)
	$p = 0.0003$	$p = 0.0003$	$p = 0.0003$	$p = 0.0003$

Table 2: Pairwise comparisons of the source vocabulary sizes 500 and 18000; 1000 and 6000. The results are calculated for the concatenated test sets generated with 8 random seeds. Inside brackets is the true mean estimated from bootstrap resampling and the 95% confidence interval. The results for the individual seeds are presented in Appendix A in Table 4 and Figure 3.

benchmarks (mainly Burlot and Yvon (2017)) that focus on a smaller number of morphological phenomena. One benefit of our method is its comprehensiveness, focusing on the corpus-wide distributions of word forms.

Using only corpus-wide metrics such as BLEU, as we used, does not discriminate between the morphological errors, which we are interested in, and other kinds of translation errors. In the terminology of Burlot and Yvon (2017), this *holistic*, document-level evaluation can be contrasted with *analytic* evaluation that focuses more specifically on difficulties in morphology. A trick that could enable a more analytic assessment of the translations of the unseen word forms would be to align the words in the source sentences with the words in the reference translations and the words in the predicted translations, and evaluate only the translations of the parts of the sentences that correspond to the unseen word forms. Similar method has been used previously for example by Bau et al. (2019); Stanovsky et al. (2019).

Especially combined with this word-alignment trick, we could also make our evaluation more *fine-grained* (this concept also from Burlot and Yvon (2017)), that is, our evaluation could differentiate between different types of mistakes. Since we have the morphological tags, we could sort the words by morphological category and compare the translation accuracies to look for any especially difficult categories for the translation models.

To demonstrate the use of our proposed method, we compared NMT models with different BPE vocabulary sizes, since vocabulary size has been hypothesised to affect the capacity to model morphology in translation. Besides vocabulary size, there are many other model design choices that have been proposed to help either in generalisation or in capturing morphological phenomena. Tokenisation methods that are more linguistically motivated than BPE, such as the Morfessor methods (Creutz and Lagus, 2002; Virpioja et al., 2013) or LMVR (Ataman et al., 2017), should help with morphological generalisation since the tokens produced by these methods approximate the linguistic morphemes more closely. Factored NMT systems (García-Martínez et al., 2016) can cover more of the target side vocabulary than subword-based NMT systems, which can also help in modelling the morphology of the target language. We hope our evaluation method will help assessing alternative NMT methods, such as these, from the perspective of morphological generalisation.

The DBCA method is general, and could be applied to a wide variety of tasks and datasets. Our application of DBCA is more specific, but it still inherits some of the generality of the original method. Our method is directly applicable to any machine learning task in which the dataset consists of sentences for which the morphological tags are available. In the future, we intend to extend our assessment of morphological generalisation to other languages, as well as to other NLP tasks, such as paraphrase detection.

6 Conclusion

We proposed a method to assess morphological generalisation by distribution-based compositionality assessment. Because this method is fully automated, it enables more comprehensive assessment of morphological generalisation than previously proposed synthetic benchmarks, in terms

of the number of inflection types we can evaluate. We used our method to assess NMT models with different BPE vocabulary sizes and found that models with smaller vocabularies are better at morphological generalisation than those with larger vocabularies. Lastly, we discussed the varied future directions that our generalisable method offers, such as assessing morphological generalisation in other NLP tasks besides NMT.

7 Acknowledgements

We thank Jörg Tiedemann, Eetu Sjöblom and others in the Helsinki Language Technology and Aalto Speech Recognition research groups for helpful discussions and advice. We also thank the anonymous reviewers for their insightful comments and feedback. The work was supported by the Academy of Finland grant 337073. The computational resources were provided by Aalto ScienceIT.

References

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from turkish to english. *The Prague Bulletin of Mathematical Linguistics*, 108(1):331–342.

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A configurable parallel corpus filtering toolbox. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 150–156. Association for Computational Linguistics.

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. 2018. Systematic generalization: What is required and can it be learned? In *International Conference on Learning Representations*.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James R. Glass. 2019. Identifying and controlling important neurons in neural machine translation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jean Berko. 1958. The child's learning of english morphology. *Word*, 14(2-3):150–177.

Franck Burlot and François Yvon. 2017. Evaluating the morphological competence of machine translation systems. In *Proceedings of the Second Conference on Machine Translation*, pages 43–55.

Leshem Choshen and Omri Abend. 2019. Automatically extracting challenge sets for non-local phenomena in neural machine translation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 291–303.

JK Chung, PL Kannappan, CT Ng, and PK Sahoo. 1989. Measures of distance between probability distributions. *Journal of mathematical analysis and applications*, 138(1):280–292.

Maria Corkery, Yevgen Matusevych, and Sharon Goldwater. 2019. Are we there yet? encoder-decoder neural networks as cognitive models of english past tense inflection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3868–3877.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2022. The paradox of the compositionality of natural language: A neural machine translation case study. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4154–4175.

Lucia Donatelli and Alexander Koller. 2023. Compositionality in computational linguistics. *Annual Review of Linguistics*, 9.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares. 2016. Factored neural machine translation architectures. In *Proceedings of the 13th International Conference on Spoken Language Translation*.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795.

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, Dennis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell, and Zhijing Jin. 2022. State-of-the-art generalisation research in NLP: a taxonomy and review. *CoRR*.

Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku neural parser pipeline: An end-to-end system for the conll 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Jordan Kodner, Salam Khalifa, Khuyagbaatar Batsuren, Hossep Dolatian, Ryan Cotterell, Faruk Akkus, Antonios Anastasopoulos, Taras Andrushko, Aryaman Arora, Nona Atanalov, et al. 2022. Sigmorphon–unimorph 2022 shared task 0: Generalization and typologically diverse morphological inflection. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 176–203.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2873–2882. PMLR.

Yafu Li, Yongjing Yin, Yulong Chen, and Yue Zhang. 2021. On compositional generalization of neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4767–4780.

Jindřich Libovickỳ and Alexander Fraser. 2020. Towards reasonably-sized character-level transformer nmt by finetuning subword systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2572–2579.

Gary F Marcus. 1998. Rethinking eliminative connectionism. *Cognitive psychology*, 37(3):243–282.

Gary F. Marcus, Steven Pinker, Michael Ullman, Michelle Hollander, T. John Rosen, Fei Xu, and Harald Clahsen. 1992. Overregularization in language acquisition. *Monographs of the Society for Research in Child Development*, 57(4):i–178.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Barbara Partee et al. 1995. Lexical semantics and compositionality. *An invitation to cognitive science*, 1:311–360.

Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.

Vikas Raunak, Vaibhav Kumar, and Florian Metze. 2019. On compositionality in neural machine translation. *arXiv preprint arXiv:1911.01497*.

David E Rumelhart and James L McClelland. 1986. On learning the past tenses of english verbs.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. *ArXiv*, abs/1904.01557.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 922–938.

Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We need to talk about random splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Jörg Tiedemann. 2020. The Tatoeba Translation Challenge – Realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. *Morfessor 2.0: Python Implementation and Extensions for Morfessor Baseline*. Aalto University publication series SCIENCE + TECHNOLOGY, 25/2013. Aalto University, Helsinki.

Aimen Zerroug, Mohit Vaishnav, Julien Colin, Sebastian Musslick, and Thomas Serre. 2022. A benchmark for compositional visual reasoning. *arXiv preprint arXiv:2206.05379*.

A Detailed results

Table 3 lists the results for the different source-side (Finnish) BPE vocabulary sizes and different compound divergence values. Table 4 includes the pairwise comparisons of vocabulary sizes 500 and 18000 and 1000 and 6000 for all random seeds. Figure 3 presents in a graph the pairwise comparisons of vocabulary sizes 500 and 18000, with all compound divergence values.

Vocab size	BLEU per compound divergence 0.0	0.25	0.5	0.75	1.0
500	27.32 (27.32 ± 0.17)	26.57 (26.57 ± 0.20)	25.36 (25.35 ± 0.17)	24.77 (24.76 ± 0.18)	25.46 (25.46 ± 0.17)
1000	27.86 (27.87 ± 0.18)	27.33 (27.33 ± 0.20)	25.87 (25.87 ± 0.18)	25.56 (25.55 ± 0.18)	25.87 (25.87 ± 0.18)
2000	27.91 (27.92 ± 0.18)	27.58 (27.58 ± 0.20)	26.07 (26.07 ± 0.18)	25.53 (25.53 ± 0.18)	25.87 (25.88 ± 0.17)
3000	28.09 (28.09 ± 0.18)	27.54 (27.54 ± 0.20)	25.98 (25.97 ± 0.17)	25.69 (25.69 ± 0.18)	25.92 (25.92 ± 0.18)
6000	28.03 (28.03 ± 0.18)	27.37 (27.36 ± 0.20)	25.98 (25.98 ± 0.18)	25.44 (25.44 ± 0.19)	25.70 (25.70 ± 0.17)
9000	27.82 (27.82 ± 0.19)	27.26 (27.26 ± 0.21)	25.73 (25.73 ± 0.17)	25.36 (25.36 ± 0.19)	25.59 (25.59 ± 0.18)
18000	27.43 (27.43 ± 0.18)	26.81 (26.81 ± 0.21)	25.36 (25.35 ± 0.17)	24.74 (24.74 ± 0.19)	25.06 (25.06 ± 0.17)
	chrF2++ per compound divergence				
500	51.01 (51.01 ± 0.14)	50.58 (50.58 ± 0.16)	49.75 (49.75 ± 0.14)	49.24 (49.24 ± 0.16)	49.19 (49.19 ± 0.14)
1000	51.53 (51.53 ± 0.14)	51.33 (51.33 ± 0.16)	50.30 (50.30 ± 0.14)	49.98 (49.98 ± 0.15)	49.59 (49.59 ± 0.14)
2000	51.54 (51.54 ± 0.14)	51.52 (51.52 ± 0.16)	50.40 (50.40 ± 0.14)	49.91 (49.91 ± 0.15)	49.68 (49.68 ± 0.14)
3000	51.68 (51.69 ± 0.14)	51.47 (51.47 ± 0.16)	50.40 (50.40 ± 0.14)	50.04 (50.04 ± 0.15)	49.62 (49.62 ± 0.14)
6000	51.66 (51.66 ± 0.14)	51.33 (51.33 ± 0.16)	50.32 (50.32 ± 0.14)	49.79 (49.79 ± 0.16)	49.48 (49.48 ± 0.14)
9000	51.37 (51.37 ± 0.14)	51.09 (51.09 ± 0.16)	50.07 (50.07 ± 0.14)	49.78 (49.77 ± 0.16)	49.36 (49.36 ± 0.14)
18000	51.02 (51.03 ± 0.14)	50.78 (50.78 ± 0.16)	49.74 (49.74 ± 0.14)	49.23 (49.23 ± 0.15)	48.78 (48.78 ± 0.14)

Table 3: The BLEU and chrF2++ results for the different source-side (Finnish) BPE vocabulary sizes and different compound divergence values. Inside brackets is the true mean estimated from bootstrap resampling and the 95% confidence interval.

Seed	Vocab	chrF2++ $\mathcal{D}_C = 0$	chrF2++ $\mathcal{D}_C = 1$	BLEU $\mathcal{D}_C = 0$	BLEU $\mathcal{D}_C = 1$
11	500	51.01 (51.01 ± 0.14)	49.19 (49.19 ± 0.14)	27.32 (27.32 ± 0.17)	25.46 (25.46 ± 0.17)
	18000	51.02 (51.03 ± 0.14)	48.78 (48.78 ± 0.14)	27.43 (27.43 ± 0.18)	25.06 (25.06 ± 0.17)
		p = 0.2439	p = 0.0003	p = 0.0243	p = 0.0003
22	500	51.01 (51.01 ± 0.14)	49.08 (49.08 ± 0.15)	27.3 (27.3 ± 0.18)	25.2 (25.2 ± 0.18)
	18000	50.85 (50.85 ± 0.14)	49.05 (49.05 ± 0.15)	27.17 (27.17 ± 0.18)	25.1 (25.1 ± 0.18)
		p = 0.0003	p = 0.1913	p = 0.0107	p = 0.053
33	500	51.07 (51.07 ± 0.14)	49.37 (49.37 ± 0.17)	27.37 (27.37 ± 0.18)	25.09 (25.09 ± 0.2)
	18000	50.97 (50.97 ± 0.14)	49.04 (49.04 ± 0.17)	27.3 (27.3 ± 0.18)	24.83 (24.83 ± 0.2)
		p = 0.0047	p = 0.0003	p = 0.092	p = 0.0003
44	500	52.02 (52.02 ± 0.17)	49.7 (49.7 ± 0.18)	28.3 (28.3 ± 0.21)	25.8 (25.8 ± 0.22)
	18000	52.44 (52.44 ± 0.17)	49.43 (49.43 ± 0.17)	28.72 (28.72 ± 0.21)	25.63 (25.63 ± 0.22)
		p = 0.0003	p = 0.0003	p = 0.0003	p = 0.0077
55	500	52.33 (52.34 ± 0.18)	49.34 (49.34 ± 0.16)	29.04 (29.04 ± 0.23)	25.29 (25.29 ± 0.2)
	18000	52.76 (52.76 ± 0.18)	49.04 (49.04 ± 0.16)	29.58 (29.58 ± 0.24)	25.08 (25.08 ± 0.2)
		p = 0.0003	p = 0.0003	p = 0.0003	p = 0.001
66	500	50.98 (50.98 ± 0.14)	49.24 (49.24 ± 0.14)	27.12 (27.12 ± 0.18)	25.31 (25.31 ± 0.18)
	18000	51.06 (51.06 ± 0.14)	48.87 (48.87 ± 0.14)	27.4 (27.4 ± 0.18)	25.04 (25.04 ± 0.17)
		p = 0.0183	p = 0.0003	p = 0.0003	p = 0.0003
77	500	50.84 (50.83 ± 0.14)	49.46 (49.46 ± 0.14)	27.12 (27.12 ± 0.18)	25.41 (25.4 ± 0.16)
	18000	50.68 (50.68 ± 0.14)	49.22 (49.22 ± 0.14)	27.06 (27.06 ± 0.18)	25.25 (25.25 ± 0.17)
		p = 0.0007	p = 0.0003	p = 0.1186	p = 0.0023
88	500	50.97 (50.97 ± 0.14)	49.38 (49.38 ± 0.14)	27.22 (27.22 ± 0.18)	25.61 (25.61 ± 0.17)
	18000	51.37 (51.37 ± 0.14)	49.05 (49.05 ± 0.14)	27.81 (27.81 ± 0.18)	25.43 (25.43 ± 0.18)
		p = 0.0003	p = 0.0003	p = 0.0003	p = 0.0003
11	1000	51.53 (51.53 ± 0.14)	49.59 (49.59 ± 0.14)	27.86 (27.87 ± 0.18)	25.87 (25.87 ± 0.18)
	6000	51.66 (51.66 ± 0.14)	49.48 (49.48 ± 0.14)	28.03 (28.03 ± 0.18)	25.7 (25.7 ± 0.17)
		p = 0.0003	p = 0.0017	p = 0.0013	p = 0.001
22	1000	51.46 (51.46 ± 0.14)	49.64 (49.64 ± 0.15)	27.9 (27.9 ± 0.18)	25.69 (25.69 ± 0.18)
	6000	51.47 (51.47 ± 0.14)	49.61 (49.61 ± 0.15)	27.94 (27.94 ± 0.19)	25.64 (25.64 ± 0.18)
		p = 0.3059	p = 0.1786	p = 0.1519	p = 0.1383
33	1000	51.59 (51.59 ± 0.14)	49.7 (49.7 ± 0.17)	27.89 (27.88 ± 0.18)	25.45 (25.45 ± 0.2)
	6000	51.63 (51.63 ± 0.14)	49.67 (49.68 ± 0.17)	28.02 (28.02 ± 0.18)	25.51 (25.51 ± 0.21)
		p = 0.117	p = 0.2073	p = 0.0047	p = 0.1276
44	1000	52.67 (52.67 ± 0.16)	50.32 (50.32 ± 0.17)	29.01 (29.01 ± 0.21)	26.53 (26.53 ± 0.22)
	6000	52.68 (52.68 ± 0.16)	50.06 (50.06 ± 0.18)	29.01 (29.01 ± 0.22)	26.33 (26.33 ± 0.22)
		p = 0.2809	p = 0.0003	p = 0.3949	p = 0.0037
55	1000	52.8 (52.8 ± 0.18)	49.92 (49.92 ± 0.16)	29.66 (29.66 ± 0.24)	25.92 (25.92 ± 0.2)
	6000	53.02 (53.03 ± 0.18)	49.72 (49.73 ± 0.16)	29.84 (29.85 ± 0.24)	25.73 (25.73 ± 0.2)
		p = 0.0003	p = 0.0003	p = 0.0017	p = 0.0003
66	1000	51.39 (51.39 ± 0.14)	49.57 (49.57 ± 0.14)	27.64 (27.64 ± 0.18)	25.71 (25.71 ± 0.18)
	6000	51.5 (51.49 ± 0.14)	49.37 (49.37 ± 0.14)	27.79 (27.79 ± 0.19)	25.54 (25.54 ± 0.18)
		p = 0.0013	p = 0.0003	p = 0.0017	p = 0.0017
77	1000	51.51 (51.51 ± 0.15)	49.8 (49.8 ± 0.13)	27.86 (27.86 ± 0.18)	25.84 (25.84 ± 0.17)
	6000	51.76 (51.76 ± 0.14)	49.74 (49.74 ± 0.14)	28.09 (28.09 ± 0.19)	25.74 (25.74 ± 0.17)
		p = 0.0003	p = 0.0453	p = 0.0003	p = 0.022
88	1000	51.9 (51.9 ± 0.14)	49.95 (49.95 ± 0.14)	28.29 (28.29 ± 0.18)	26.2 (26.2 ± 0.18)
	6000	51.6 (51.6 ± 0.14)	49.84 (49.84 ± 0.14)	28.01 (28.01 ± 0.18)	26.23 (26.23 ± 0.18)
		p = 0.0003	p = 0.0007	p = 0.0003	p = 0.2209

Table 4: Pairwise comparisons of the source vocabulary sizes 500 and 18000; 1000 and 6000 on the minimum and maximum compound divergence data splits. For 8 data split algorithm random seeds. Inside brackets is the true mean estimated from bootstrap resampling and the 95% confidence interval.

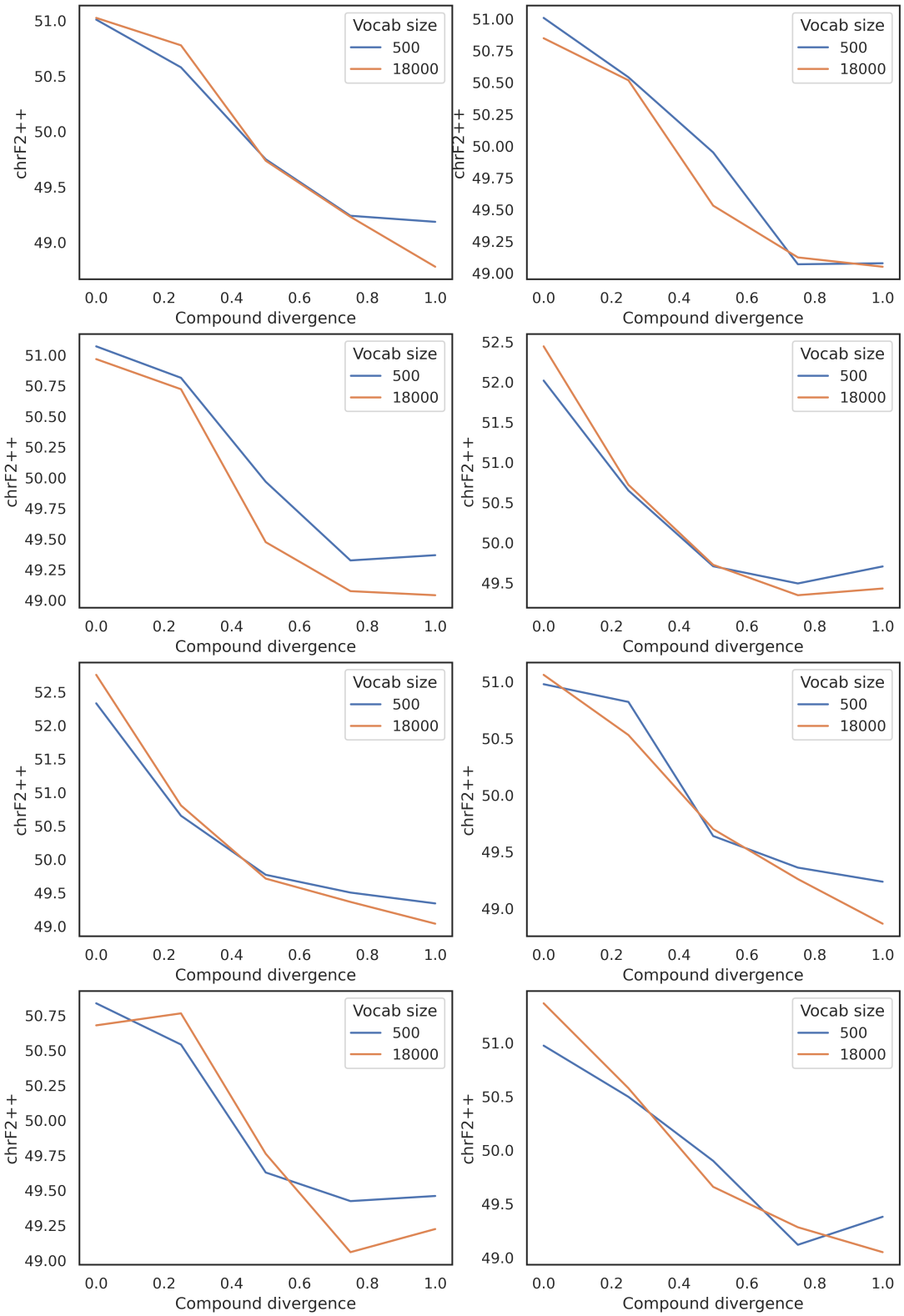


Figure 3: Comparison of vocabulary sizes 500 and 18000 with compound divergence values 0.0, 0.25, 0.5, 0.75 and 1.0. For 8 data split algorithm random seeds. The same results are partly in Table 4.

Estonian Named Entity Recognition: New Datasets and Models

Kairit Sirts
Institute of Computer Science
University of Tartu
sirts@ut.ee

Abstract

This paper presents the annotation process of two Estonian named entity recognition (NER) datasets, involving the creation of annotation guidelines for labeling eleven different types of entities. In addition to the commonly annotated entities such as person names, organization names, and locations, the annotation scheme encompasses geopolitical entities, product names, titles/roles, events, dates, times, monetary values, and percents. The annotation was performed on two datasets, one involving reannotating an existing NER dataset primarily composed of news texts and the other incorporating new texts from news and social media domains. Transformer-based models were trained on these annotated datasets to establish baseline predictive performance. Our findings indicate that the best results were achieved by training a single model on the combined dataset, suggesting that the domain differences between the datasets are relatively small.

1 Introduction

Named entity recognition (NER) is a practical natural language processing (NLP) task that involves identifying and extracting named entities from texts, such as person names, organization names, locations, and other types of entities. NER is widely used in various downstream applications, such as document anonymisation and text categorisation. Typically, modern NER systems are trained as supervised tagging models, where annotated training data is utilised for training models to identify and tag text spans that correspond to named entities.

For the Estonian language, prior endeavors to develop NER systems have involved the creation of an annotated dataset labelled with person, organisation, and location names (Tkachenko et al., 2013). This dataset has been utilised for training CRF- and transformer-based NER models (Tkachenko et al., 2013; Kittask et al., 2020; Tanvir et al., 2021). In addition to these efforts, a dataset in a different domain, 19th-century parish court records, was recently annotated with named entities (Orasmaa et al., 2022).

This paper describes the efforts to augment further the development of general-purpose named NER systems for the Estonian language. The primary focus of this study is annotating additional Estonian texts with named entities, utilising a newly developed rich annotation scheme. Two annotated datasets were created as part of this effort. Firstly, the existing NER dataset (Tkachenko et al., 2013) was reannotated using the new annotation scheme. Secondly, approximately 130K tokens of new texts, predominantly sourced from news portals and social media, were annotated to create a new dataset. These annotations serve to expand the availability of annotated data for training and evaluating NER models in the Estonian language.

The second part of this paper delves into the experimental results obtained from training predictive BERT-based models on the annotated datasets. The primary objectives of these experiments were to establish the baseline performance of various entity types of the newly developed annotation scheme and to explore the optimal utilisation of the two datasets, which stem from slightly distinct domains. The findings revealed that the baseline performance on the newly annotated dataset was slightly lower than the less richly annotated Estonian NER dataset, indicating that the new annotations may possess some noise while also being richer and more intricate. Moreover, the study revealed that the domains of the two datasets were similar enough such that a model

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 752–761

trained on the combined dataset exhibited comparable or even superior performance compared to models trained on each dataset separately.

In short, our paper makes two key contributions:

1. The introduction of two novel Estonian NER datasets that are annotated with a comprehensive set of entities, enriching the available resources for NER research in Estonian;

2. An evaluation of the performance of BERT-based models on the newly annotated datasets, providing baseline assessments for these datasets.

2 Dataset Creation

This section describes the process of creating the two labelled NER datasets for Estonian.[1]

2.1 Data Sources

The first dataset, referred to as the Main NER dataset in our study, is a reannotation of the existing Estonian NER dataset (Tkachenko et al., 2013). This dataset comprises approximately 220K words of news texts and exhibits a homogeneous domain. Notably, previous studies have identified errors in the annotations of this dataset (Tanvir et al., 2021), which motivated us to undertake its reannotation.

The second dataset, referred to as the New NER dataset in our study, is newly created. We aimed to select approximately 130K tokens from news and social media domains, with around 100K tokens from the news domain and 30K tokens from the social media domain. To obtain the texts, we sampled from the Estonian Web Corpus 2017 (Jakubíček et al., 2013), utilizing metadata such as URL and web page title for text selection. For news sources, we identified URLs and titles associated with major Estonian news sites such as *Postimees*, *EPL*, *ERR*, and *Delfi*. For social media texts, we searched for keywords indicative of well-known blogging and forum platforms such as *blogspot* and *foorum*.

2.2 Annotation Guidelines

We devised annotation guidelines to label the data, aiming to adopt a more comprehensive set of labels beyond the commonly used person, organisation, and location names.[2] We decided to differentiate between geopolitical entities and geographical locations. Following similar works in Finnish (Ruokolainen et al., 2020), we introduced labels for events, products, and dates. Furthermore, we included titles, times, monetary values, and percentages. The annotation guidelines included a brief description for each entity, as used during the annotation process, which was as follows:

- Persons (PER): This includes names referring to all kinds of real and fictional persons.

- Organizations (ORG): This includes all kinds of clearly and unambiguously identifiable organizations, for example, companies and similar commercial institutions as well as administrative bodies.

- Locations (LOC): This includes all geographical locations not associated with a specific political organization such as GPEs.

- Geopolitical entities (GPE): This includes all geographic locations associated with a political organization, such as countries, cities, and empires.

- Titles (TITLE): This includes job titles, positions, scientific degrees, etc. Only those titles should be annotated where a specific person behind the title can be identified based on the preceding text. The personal name immediately following the title is not part of the TITLE. If the ORG tag precedes the title, only the job title must be marked with the TITLE, not the words in the ORG.

- Products (PROD): This includes all identifiable products, objects, works, etc., by name.

- Events (EVENT): This includes events with a specific name.

- Dates (DATE): This includes time expressions, both in day/month/year type, e.g., "October 3rd", "in 2020", "2019", "in September", as well as general expressions ("yesterday", "last month", "next year") if the expression has a clear referent. The criterion is that based on the expression, it must be possible to determine a specific point in time, i.e., a

[1] The annotated datasets are available:
https://github.com/TartuNLP/EstNER
https://github.com/TartuNLP/EstNER_new

[2] The annotation guidelines in Estonian are available upon request.

specific year, month, or day. Thus, vague expressions such as ”a few years from now“, ”a few months ago“ are not suitable, but more specific expressions such as ”five years later“, ”three months ago“, or ”the day before yesterday“ are suitable.

- Times (TIME): This includes time expressions that refer to an entity smaller than a day: times and parts of a day with a referent (analogous to DATE entities). General expressions without a referent are not marked. Durations are also not marked.

- Monetary values (MONEY): This includes expressions that refer to specific currencies and amounts in those currencies.

- Percentages (PERCENT): This includes entities expressing percentages. A percentage can be expressed both with a percentage mark (%) or verbally.

2.3 Nested Entities

Similar to Ruokolainen et al. (2020), we incorporated nested entities into our annotation schema. For instance, an example of a nested entity would be ”New York City Government“, where the ORG entity ORG encompasses the nested GPE entity ”New York“. We set a limit of up to three levels of nesting. However, we restricted the annotation of nested entities of the same type, except for ORG. For instance, if ”The Republic of Ireland“ was annotated as GPE, further annotation of ”Ireland“ as a nested GPE was not permitted. Nevertheless, in cases such as ”The UN Department of Economic and Social Affairs“ labelled as ORG, the word token ”The UN“ would be allowed to be annotated as a nested ORG.

2.4 Annotation Process

The process of annotation was carried out separately for both datasets. For the Main NER dataset, three annotators, who were graduate students in general or computational linguistics, were recruited. All annotators were native speakers of Estonian. Each annotator independently labelled the dataset based on the provided guidelines. Two annotators completed annotations for the entire dataset, while one annotator completed most of the annotations, with a few documents remaining. The annotation of the Main NER dataset was conducted using Label Studio, a freely available open-source platform for data annotation.

A total of twelve annotators were involved in annotating the New NER dataset. Two annotators completed the entire annotation process. One of them was an undergraduate linguistic student, while the other was a graduate student in computer science with an undergraduate degree in linguistics. The remaining ten annotators participated in a graduate-level NLP course, and each annotated approximately 12K word tokens as part of their coursework. All annotators were native speakers of Estonian. All annotators worked independently, without access to each other's work, adhering to the provided annotation guidelines. As a result, each text in the New NER dataset received three independent annotations. The annotation of the New NER dataset was performed using DataTurks, an annotation platform currently non-existent.

2.5 Label Harmonisation

Harmonising the annotations in the New NER dataset involved both automatic and manual approaches. Initially, automatic harmonisation was applied based on the following principle. If annotators A and B had agreed on a particular annotation, but annotator C had not provided any annotation, the final label was set to the annotation agreed upon by A and B. Subsequently, the entire corpus was manually reviewed by two individuals, one of whom was the original annotator A, and the other was the author of this paper. Through discussion and deliberation, the labels were disambiguated. In most cases, the final label chosen was the one that at least two annotators had selected. However, in some instances, the label was changed entirely, or a completely new span of words was annotated as an entity based on mutual agreement.

The disambiguation of annotations in the Main NER dataset was carried out automatically. As per the automatic procedure, a word span was labelled as an entity if it had been marked as such by at least two annotators and they had used the same tag for that entity.

2.6 Inter-Annotator Agreement

In order to evaluate the reliability of the annotations, inter-annotator agreements were computed for the Main NER dataset, as shown in Table 1. Fleiss' kappa, an extension of Cohen's kappa to

	1st level	2nd level	3rd level
	0.65	0.23	-0.16
PER	0.95	0.27	0.66
ORG	0.76	0.33	0.19
LOC	0.65	0.35	0.18
GPE	0.84	0.47	-0.08
TITLE	0.63	0.21	0.00
PROD	0.48	0.02	–
EVENT	0.43	0.53	–
DATE	0.72	0.06	–
TIME	0.53	0.00	–
MONEY	0.78	0.00	–
PERCENT	0.90	–	–

Table 1: Inter-annotator agreement of the Main NER dataset, measured with the Fleiss κ.

accommodate more than two annotators, was computed following the procedure outlined by Ruokolainen et al. (2020). Each entity occurrence in the text was treated as an instance of the positive class, and the exact match of annotations between annotators was checked for each entity. If annotators had marked the same entity with the same label, it was recorded as an instance of the positive class; otherwise, it was recorded as an instance of the negative class.

The inter-annotator agreement for the 1st level entities was found to be in the range of substantial agreement. However, in contrast, the annotations for the second and third levels showed lower agreement, as indicated by Fleiss' kappa's low or even negative values. Specifically, person names, geopolitical entities, and percentages achieved almost perfect agreement ($\kappa > 0.8$) at the first level. Most other entity types showed substantial agreement ($\kappa > 0.6$). The lowest agreement scores were observed for products and events, which still obtained moderate agreement ($\kappa > 0.4$).

2.7 Final Datasets

Following the label harmonisation process, the resulting datasets were divided into the train, validation, and test splits. These datasets and the prepared splits will be made available for future comparisons of developed models. Table 2 presents the final datasets' statistics.

The Main NER dataset was previously annotated with only three entity types: PER, ORG, and LOC, as reported by Tkachenko et al. (2013). Among these, PER and ORG labels remain the most frequently occurring ones in the dataset. However, there have been changes in the annotation guidelines, resulting in most LOC annotations being replaced with GPE. Additionally, the Main NER dataset contains a relatively large number of titles, dates, and products. On the other hand, the occurrence of event entities is comparatively low in this dataset.

Similar trends in entity prevalence can be observed in the New NER dataset. PER, ORG, and GPE entities remain the most frequent, followed by a relatively large number of titles, dates, and products. Notably, the New NER dataset contains a higher occurrence of EVENT entities compared to the Main NER dataset. However, TIME, PERCENT, and MONEY entities are less frequent in the New NER dataset.

3 Experiments

We had two primary goals when conducting the experiments. The first goal was to establish the baseline performance on both the Main NER and New NER datasets. While several previous studies have reported results on the old annotations of the Main NER dataset, the new annotations we used in our study are more comprehensive and were collected independently without reference to the old annotations. Therefore, the baseline performance of the Main NER dataset with the new annotations may differ. Similarly, as the New NER dataset contains new material, it is crucial to evaluate its baseline performance as well.

The second goal of our study was to investigate potential domain differences between the two datasets. Specifically, the average document length in the New NER dataset was more than three times higher than that of the Main NER dataset. Also, the New NER dataset contains at least 30K tokens from the social media domain. Moreover, the news part of the New NER dataset documents was not limited to formal news texts but also included less formal opinion pieces. Hence, our objective was to determine the optimal approach for utilising these datasets, namely whether training separate models for each dataset would be more effective or if combining the data and training a single model would yield better results.

We opted to utilise only the first-level annotations for training our models. This decision was

	Main NER dataset				New NER dataset			
	Train	**Val**	**Test**	**Total**	**Train**	**Val**	**Test**	**Total**
Documents	525	18	39	582	78	16	15	109
Sentences	9965	2415	1907	14287	7001	882	890	8773
Tokens	155983	32890	28370	217243	111858	13130	14686	139674
1st lvl entities	14944	2808	2522	20274	8078	541	1002	9594
2nd lvl entities	987	223	122	1332	571	44	59	674
3rd lvl entities	40	14	4	58	27	0	1	28
PER	3563	642	722	4927	2601	109	299	3009
ORG	3215	504	541	4260	1177	85	150	1412
LOC	328	118	61	507	449	31	35	515
GPE	3377	714	479	4570	1253	129	231	1613
TITLE	1302	171	209	1682	702	19	59	772
PROD	874	161	66	1101	624	60	117	801
EVENT	56	13	17	86	230	15	26	271
DATE	1346	308	186	1840	746	64	77	887
TIME	456	39	30	525	103	6	6	115
PERCENT	137	62	58	257	75	11	1	87
MONEY	291	76	153	520	118	12	1	131

Table 2: Statistics of the two new Estonian NER datasets.

based on the finding that much fewer entities were labelled at the second and third levels, as evidenced by the statistics presented in Table 2. Furthermore, the inter-annotator agreements for the second and third-level entities were found to be lacking, as illustrated in Table 1. Hence, we focused solely on the first-level annotations to ensure a more reliable and consistent training process.

4 Model

We employed a transformer-based token classification model for our experiments, adopting the commonly-used BIO format for entity labelling. In this format, the B-tag indicates the start of an entity, the I-tag denotes the continuation of an entity, and the O-tag is assigned to word tokens that do not belong to any named entity. The TokenClassification implementation from the Huggingface `transformers` library (Wolf et al., 2020) was utilised for this purpose. As our base model, we used the EstBERT model with a sequence length of 128[3] (Tanvir et al., 2021), which was fine-tuned on the NER datasets.

In our experiments, we kept the batch size fixed at 16 and utilised the Adam optimiser with betas set to 0.9 and 0.98 and an epsilon value of 1e-6. The models were trained for a maximum of 150 epochs, with early stopping implemented if the overall F1-score on the validation set did not improve for 20 consecutive epochs by more than 0.0001 F1-score points. We used the `seqeval` package (Nakayama, 2018) for evaluations during training and final testing. The learning rate was optimised on the validation set using a grid of values 5e-6, 1e-5, 3e-5, 5e-5, 1e-4. Each model was trained ten times with different random seeds to account for randomness, and the mean values with standard deviations are reported.

5 Results

We first trained and evaluated models separately on both datasets to assess their overall modeling performance. Then, we trained a joint model using data from both datasets and compared its performance on the evaluation sets of both datasets. This allowed us to evaluate the effectiveness of using a combined dataset compared to training on each dataset separately.

[3] `https://huggingface.co/tartuNLP/EstBERT`

	Reannoated Main NER				New NER			
	#	Precision	Recall	F1-score	#	Precision	Recall	F1-score
PER	642	.827 (.012)	.871 (.009)	.848 (.005)	109	.809 (.044)	.816 (.023)	.811 (.019)
ORG	504	.654 (.016)	.666 (.014)	.660 (.013)	85	.580 (.027)	.585 (.052)	.581 (.024)
LOC	118	.643 (.036)	.478 (.028)	.547 (.016)	31	.600 (.065)	.560 (.060)	.576 (.044)
GPE	714	.821 (.012)	.831 (.021)	.826 (.008)	129	.900 (.017)	.879 (.030)	.889 (.014)
TITLE	171	.676 (.023)	.814 (.014)	.739 (.011)	19	.750 (.062)	.718 (.064)	.731 (.048)
PROD	161	.572 (.033)	.628 (.026)	.598 (.024)	60	.509 (.043)	.474 (.052)	.488 (.029)
EVENT	13	.069 (.029)	.077 (.034)	.072 (.031)	16	.518 (.104)	.558 (.104)	.525 (.070)
DATE	308	.682 (.020)	.720 (.017)	.700 (.007)	64	.816 (.027)	.824 (.024)	.820 (.021)
TIME	39	.553 (.066)	.555 (.045)	.553 (.053)	6	.812 (.041)	.788 (.108)	.797 (.074)
PERCENT	62	.985 (.016)	.867 (.032)	.922 (.019)	11	.895 (.126)	1 (–)	.940 (.074)
MONEY	76	.636 (.040)	.568 (.030)	.600 (.030)	12	.659 (.085)	.742 (.126)	.693 (.083)
Overall	2571	.737 (.010)	.757 (.009)	.747 (.004)	497	.736 (.014)	.734 (.017)	.735 (.006)

Table 3: Predictive performance of models trained on both two datasets, evaluated on the respective validation set.

5.1 Separate Models

The results of the experiments with separate models, evaluated on the respective validation sets, are reported in Table 3. The overall performance, as indicated in the bottom row of the table, is similar for both datasets, suggesting that the annotation and modeling difficulty is comparable in the two datasets.

The entities that were most accurately predicted in both datasets are PER, GPE, and PERCENT. Conversely, the lowest accuracy was observed when predicting LOC, EVENT, and TIME for the reannotated Main NER dataset, and LOC, EVENT, and PROD for the New NER dataset. Predicting EVENT names is particularly challenging in the Main NER dataset, likely due to the limited number of instances (only 56) in the respective training set.

	Precision	Recall	F1-score
PER	.948	.958	.953
ORG	.784	.826	.805
LOC	.899	.914	.907
Overall	.891	.912	.901

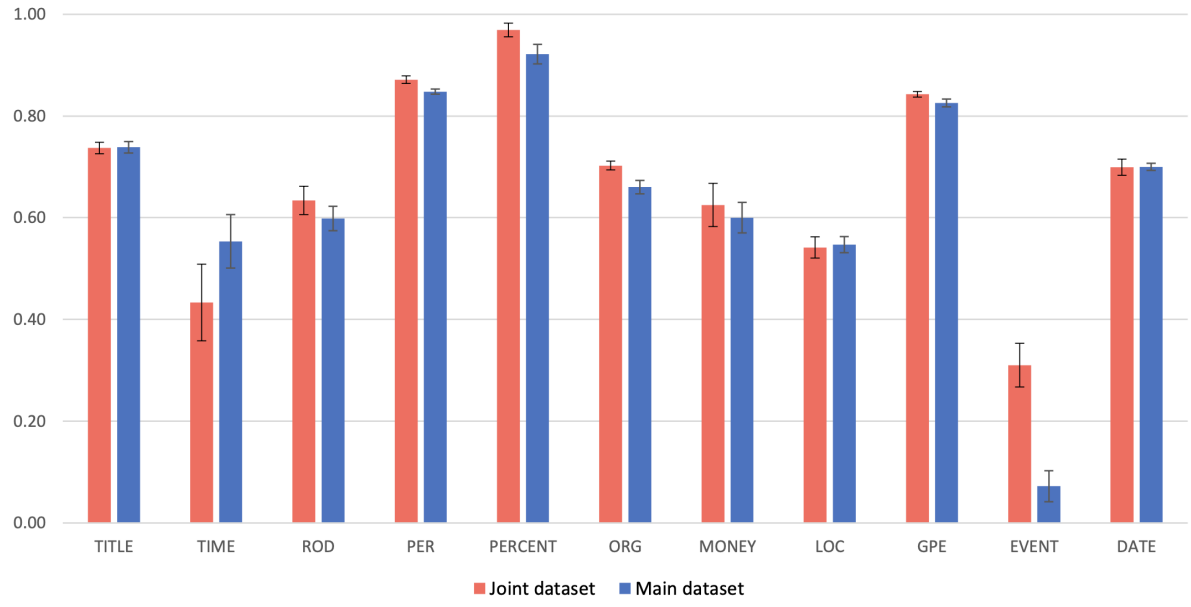
Table 4: Results of the old annotations of the Main NER test set. Adapted from Table 11 (Tanvir et al., 2021).

A comparison of the results between the Reannotated Main dataset and the previous annotations of the Main NER dataset (refer to Table 4, sourced from Tanvir et al. (2021), Table 11) reveals that the performance on all three entities (PER, ORG, LOC) used in the old annotations has declined. It should be noted that the modeling results are not directly comparable, as Table 3 presents validation set results while Table 4 presents test set results. However, the differences in performance suggest that the new annotation might be more complex for the models to learn.

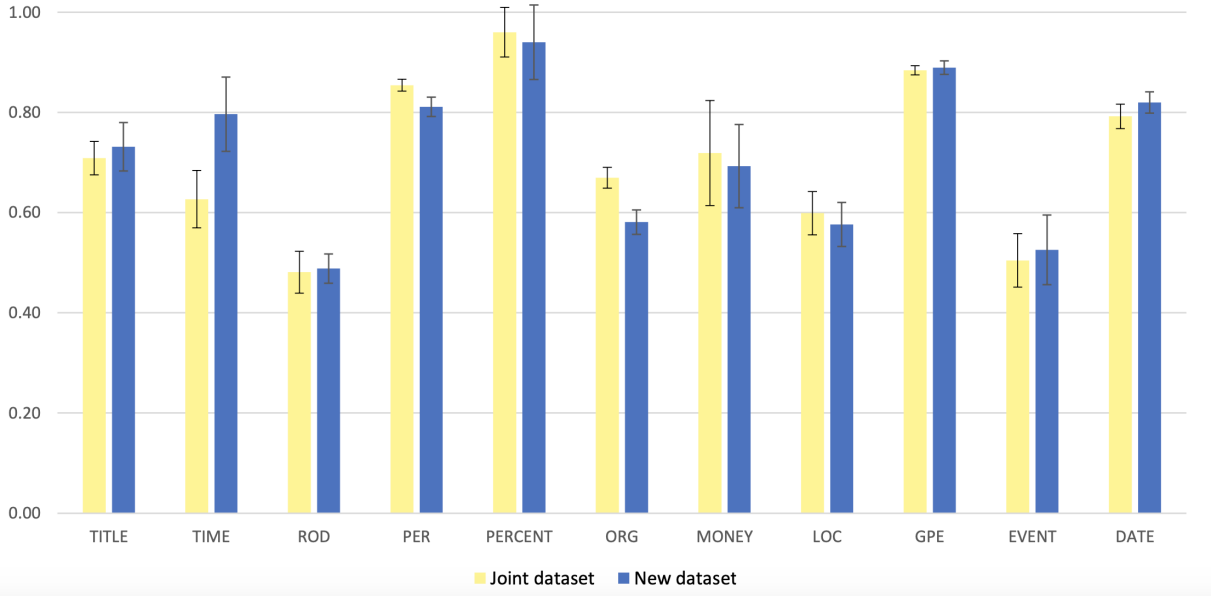
5.2 Joint Model

The joint model is trained using the combined train sets of the Main NER and New NER datasets. Table 5 presents the F1-scores of the joint model on the merged validation set and on the validation sets of both datasets individually. Notably, the overall F1-scores of the joint model are slightly higher than the F1-scores of the separate models (0.766 vs. 0.747 for the Main dataset and 0.752 vs. 0.735 for the New dataset), as evident from the bottom row of Table 3.

Figure 1 presents a detailed entity-level comparison of the joint and separate models on their respective validation sets. Specifically, Figure 1a illustrates the comparison on the validation set of the Main NER dataset. The results reveal that the joint model performs similarly or better than the separate models across most entities, except for the TIME entity, which already had low performance in the Main dataset and further decreases with the joint model from 0.553 to 0.433. Con-



(a) Evaluation on the Main NER validation set.



(b) Evaluation on the New NER validation set.

Figure 1: An entity-level comparison of the joint model against models trained on each dataset separately.

	Main+New Val F1	Main Val F1	New Val F1	Main+New Test F1	Main+New Test Prec	Rec	F1
PER	.868 (.007)	.872 (.008)	.854 (.012)	.879 (.007)	.840	.927	.882
ORG	.690 (.010)	.702 (.009)	.669 (.021)	.700 (.016)	.698	.693	.696
LOC	.549 (.019)	.541 (.021)	.599 (.043)	.526 (.025)	.478	.563	.517
GPE	.849 (.005)	.843 (.005)	.884 (.009)	.826 (.004)	.827	.830	.828
TITLE	.733 (.013)	.737 (.011)	.709 (.034)	.777 (.017)	.788	.758	.773
PROD	.598 (.018)	.634 (.028)	.481 (.042)	.568 (.020)	.576	.579	.578
EVENT	.370 (.053)	.310 (.043)	.504 (.053)	.264 (.034)	.306	.256	.278
DATE	.708 (.013)	.699 (.016)	.792 (.024)	.740 (.010)	.727	.768	.747
TIME	.451 (.065)	.433 (.075)	.627 (.057)	.463 (.043)	.548	.472	.507
PERCENT	.969 (.019)	.969 (.013)	.960 (.049)	.958 (.013)	.967	.983	.975
MONEY	.622 (.032)	.625 (.042)	.719 (.105)	.699 (.014)	.789	.614	.690
Overall	.761 (.004)	.766 (.002)	.752 (.010)	.773 (.006)	.766	.783	.774

Table 5: Evaluations of the joint model trained on the combined train sets of both datasets. Left block: F1-scores on the different portions of the validation sets. Middle block: F1-scores on the combined test set. Right block: test scores of the best-performing joint model.

versely, the prediction accuracy of the EVENT entity, while remaining relatively low, notably improves from 0.072 to 0.310 with the joint model.

Upon comparing the results of the joint and separate models on the New NER dataset (refer to Figure 1b), we observe that the joint model performs similarly or better on certain entity types, including PER, ORG, GPE, LOC, PROD, PERCENT, and MONEY while exhibiting slightly lower performance on the remaining entities. Notably, the TIME entity experiences the most significant drop in performance, declining from 0.797 to 0.627 with the joint model.

In summary, our findings support using a joint model instead of two separate models. While there may be a slight drop in prediction performance for certain entities, particularly in the New NER dataset, the overall F1-score on the validation sets of both datasets is higher with the joint model compared to the separate models. As a result, we proceed with the joint model for the final evaluations on the test set.

5.3 Test Results

The test results of the joint model on the combined test set can be found in the fourth column of Table 5. The overall F1-score is slightly higher on the test set than on the validation set. Specifically, for certain entities such as PER, ORG, TITLE, DATE, TIME, and MONEY, the test F1-score is higher than the validation F1-score, while it is slightly lower for others. Notably, the EVENT entity experiences the most significant drop in performance, with the test F1-score declining from 0.370 to 0.264.

All the results mentioned above were presented as averages across ten different runs. Additionally, we selected a joint model with the highest overall validation F1-score to make it publicly available. The test scores of this chosen model are provided in the right-most block of Table 5. The overall F1-score of this best model is in line with the mean F1-score, indicating that it was not the model with the highest F1-score on the test set. However, due to the small standard deviations observed, the results of all models are within a close range; the highest F1-score achieved on the test set is 0.785.[4]

6 Discussion

This study marks the first endeavor to annotate a more comprehensive set of entities beyond the commonly annotated person, organization, and location names in the Estonian language. The inter-annotator agreement results indicate that the annotators consistently labelled certain entities, such as PER, GPE, and PERCENT, while the reliability was lower for other entities. In particular, the EVENT entity had the lowest inter-annotator agreement. An in-depth analysis of inconsistencies in annotation, both in EVENT and other en-

[4]The best joint model is available: `https://huggingface.co/tartuNLP/EstBERT_NER_v2`

tities, could be conducted as a follow-up work to identify the sources of confusion and enhance the annotation guidelines.

In line with previous efforts in other languages, such as Finnish, we opted to annotate nested entities by permitting up to three levels of nesting. However, upon analysing the data statistics, it was revealed that only a few entities were annotated on the third level. Additionally, even though many entities were labelled on the second level, their reliability, as evidenced by inter-annotator agreements, was not deemed sufficiently high. Hence, utilising these labels for training predictive models may not yield productive results.

In this study, we obtained three sets of annotations for both datasets, enabling us to assess the variability in the annotations. However, it is essential to acknowledge that the choice of annotators may have introduced limitations to the annotation process. For the Main NER dataset, all annotators were linguistic students, which provided expertise and interest in the annotation task, as intended. However, this uniformity in the background may have resulted in limitations in the recall of entity annotations, as noted in previous research (Derczynski et al., 2016). On the other hand, the annotators for the New NER dataset were more diverse, including computer science students. Nevertheless, since the task was part of their coursework, their motivation and interest in the annotation task might not have been as high.

Our experimental results with the BERT-based model indicate that although there may be a domain shift between the two datasets at the entity level for certain entities, training a single joint model on both datasets seems justified. It is important to note that our models based on EstBERT are only baselines, and as demonstrated in previous studies (Kittask et al., 2020; Tanvir et al., 2021), utilising other base models such as Estonian WikiBERT (Pyysalo et al., 2021) or XLM-RoBERTa could potentially yield higher performance results.

7 Conclusions

We provided a detailed overview of the annotation process for two Estonian NER datasets, annotated with a comprehensive annotation scheme encompassing eleven distinct entity types. Additionally, the datasets included nested annotations of up to three levels, although the reliability of the nested annotations was found to be less consistent compared to the first-level entities. In order to establish baseline predictive accuracy, we conducted experiments with two modeling scenarios on these newly annotated datasets. This involved training two separate models, one for each dataset and a joint model on the combined dataset. Our findings revealed that the joint model outperformed the separate models, except for a few entity types, indicating that the domain differences between the datasets are relatively minimal. As such, we recommend utilising these two datasets jointly as a single, more diverse dataset for NER training purposes.

Acknowledgments

We thank Laura-Katrin Leman, Chenghan Chung and Claudia Kittask for their contributions in this work, and all data annotators. This research was supported by the Estonian Research Council Grant PSG721 and by the Estonian Language Technology Grant EKTB11.

References

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad twitter corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179.

Miloš Jakubíček, Adam Kilgarriff, Vojtěch Kovář, Pavel Rychlỳ, and Vít Suchomel. 2013. The tenten corpus family. In *7th international corpus linguistics conference CL*, pages 125–127. Lancaster University.

Claudia Kittask, Kirill Milintsevich, and Kairit Sirts. 2020. Evaluating multilingual bert for estonian. In *Baltic HLT*, pages 19–26.

Hiroki Nakayama. 2018. https://github.com/chakki-works/seqeval Seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Siim Orasmaa, Kadri Muischnek, Kristjan Poska, and Anna Edela. 2022. Named entity recognition in estonian 19th century parish court records. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5304–5313.

Sampo Pyysalo, Jenna Kanerva, Antti Virtanen, and Filip Ginter. 2021. Wikibert models: Deep transfer learning for many languages. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 1–10.

Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. 2020. A finnish news corpus for named entity recognition. *Language Resources and Evaluation*, 54:247–272.

Hasan Tanvir, Claudia Kittask, Sandra Eiche, and Kairit Sirts. 2021. Estbert: A pretrained language-specific bert for estonian. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 11–19.

Alexander Tkachenko, Timo Petmanson, and Sven Laur. 2013. Named entity recognition in estonian. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 78–83.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Machine Translation for Low-resource Finno-Ugric Languages

Lisa Yankovskaya Maali Tars Andre Tättar Mark Fishel
Institute of Computer Science
University of Tartu, Estonia
{lisa.yankovskaya,maali.tars,andre.tattar,mark.fishel}@ut.ee

Abstract

This paper focuses on neural machine translation (NMT) for low-resource Finno-Ugric languages. Our contributions are three-fold: (1) we extend existing and collect new parallel and monolingual corpora for 20 Finno-Ugric languages, (2) we expand the 200-language translation benchmark FLORES-200 with manual translations into nine new languages, and (3) we present experiments using the collected data to create NMT systems for the included languages and investigate the impact of back-translation data on the NMT performance for low-resource languages. Experimental results show that carefully selected back-translation directions in a multilingual setting yield the best results in terms of translation scores, for both high-resource and low-resource output languages.

1 Introduction

Neural networks have caused rapid growth in output quality for many natural language processing tasks, including neural machine translation (NMT, Vaswani et al., 2017). However, the output quality crucially depends on the availability of large amounts of parallel and monolingual data for the covered languages.

Recently synthetic data and cross-lingual transfer have not only shown potential for low-resource language NMT but also have been taken to the extreme through open massively multilingual translation models (Fan et al., 2021; NLLB Team et al., 2022). In addition to translation models, a massive translation benchmark FLORES-200 (NLLB Team et al., 2022) has been created, consisting of multi-parallel translations of the same sentences into 200 languages.

Here we focus on NMT for low-resource languages from a family of languages spoken in Europe, but not part of the Indo-European family: Finno-Ugric languages. Three members of that family (Estonian, Finnish and Hungarian) are commonly included in massively multilingual efforts and can be considered medium-resource languages. At the same time, several lower-resource Finno-Ugric languages are not included in the existing massively multilingual models (M2M-100, NLLB). In terms of the number of speakers, they range from 20 near-native speakers of Livonian to several hundred thousand speakers of Mordvinic languages.

Our contributions are three-fold. First, we present a collection of parallel and monolingual corpora that can be used for training NMT systems for 20 low-resource Finno-Ugric languages. The resources are collected from sources that are already digital (primarily online sources); the languages and the data are described in Section 3.

Secondly, we expand a part of the 200-language translation benchmark FLORES-200 with manual translations into the low-resource Finno-Ugric languages. This includes the first 250 sentences of FLORES-200 and the following languages: Komi, Udmurt, Hill and Meadow Mari, Erzya, Livonian, Mansi, Moksha and Livvi Karelian. This new benchmark is described in Section 4.

Finally, we use the collected parallel and monolingual data in experiments to create NMT systems for the covered languages. The main question we address is which subsets of translation directions yield the best results for the included low-resource languages. We achieve an average chrF++ score of 26.8 when translating from high-resource to low-resource languages included in our expansion of FLORES-200. The complete experiments and results are presented in Sections 5 and 6.

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 762–771

2 Related Work

Low-resource NMT Machine translation is dominated by neural methods in current research. Neural machine translation also requires large amounts of training segments for high-quality translation across different domains. That is a challenge when it comes to low-resource languages.

In Gu et al. (2018) and Sennrich and Zhang (2019), the authors investigate the best NMT model setups, with Sennrich and Zhang (2019) showing a comparison to phrase-based systems that are not that common these days. Gu et al. (2018) and Kocmi and Bojar (2018) indicate that training universal models (sharing parameters between multiple languages) and transfer learning are two aspects that get significant gains for low-resource language pairs in translation quality.

More recently, low-resource machine translation has risen to the attention of more and more research groups with multiple comprehensive surveys emerging (Haddow et al., 2022; Wang et al., 2021), showing that there has already been a lot of work done that can now be systematically aggregated and utilized in further research.

Low-resource NMT for Finno-Ugric languages Some of the Finno-Ugric languages have been considered in the context of NMT before. Tars et al. (2021, 2022a,b) and Rikters et al. (2022) present experiments with several Sami languages, Võro and Livonian. They used similar techniques like multilinguality, pre-trained models, transfer learning, and back-translation to better the translation quality. Our work aims to bridge the gap between the other low-resource Finno-Ugric languages and those that already have good support, offered by the previously published papers.

In 2022, Livonian-English was part of the translation shared task at WMT, the International Conference of Machine Translation (Kocmi et al., 2022). A Livonian-English test set was created; in our work, we add Livonian to FLORES-200, which covers several language pairs more than the WMT'22 test set.

Back-translation in low-resource setting Back-translation is a widely used method for enhancing translation quality while making use of monolingual data (Sennrich et al., 2016). This is also one of the aspects that allows for good quality NMT systems in the low-resource setting because low-resource languages lack parallel data while monolingual data is often much easier to find.

There has been research into exploring the specifics of back-translation like models used for synthetic data creation, beam search vs greedy search, the domain of monolingual data as well as amounts of synthetic data (Edunov et al., 2018), the last of them is the closest we also desire to investigate in our low-resource Finno-Ugric setting. Other research goes into detail about how diverse the synthetic data should be (Burchell et al., 2022) and how effective iterative back-translation is (Hoang et al., 2018).

Pre-trained models For multilingual NMT, it has become insufficient to train models from scratch, instead using pre-trained models has become a prevalent method for all NLP tasks. In machine translation, the massively multilingual models of M2M-100 and NLLB are a good starting point to use for fine-tuning and transfer learning (Fan et al., 2021; NLLB Team et al., 2022).

3 Languages and Data

The Finno-Ugric language group has two major branches: Finno-Permic and Ugric. Although both branches share common linguistic roots, they are quite distant.

The Finno-Permic branch includes two high-resource languages, Estonian and Finnish, and several low-resource languages, such as Komi, Komi Permyak, Udmurt, Hill and Meadow Mari, Erzya and Moksha, Proper and Livvi Karelian, Ludian, Võro, Veps, Livonian, Sami languages. The Ugric branch comprises three languages: high-resource Hungarian and two low-resource Mansi and Khanty.

In this work, we develop an NMT system between 20 low-resource Finno-Ugric (FU) languages shown in Figure 1 and seven high-resource languages (English, Estonian, Finnish, Hungarian, Latvian, Norwegian (Bokmål), and Russian). The selection of the high-resource languages is not accidental: Estonian, Finnish, and Hungarian belong to the FU language family, while Latvian has markedly influenced Livonian, Norwegian has deeply affected the Sami languages and Russian has had a profound impact on the Permic, Mordvinic, Mari, Karelian, Veps, and Ob-Ugric languages.

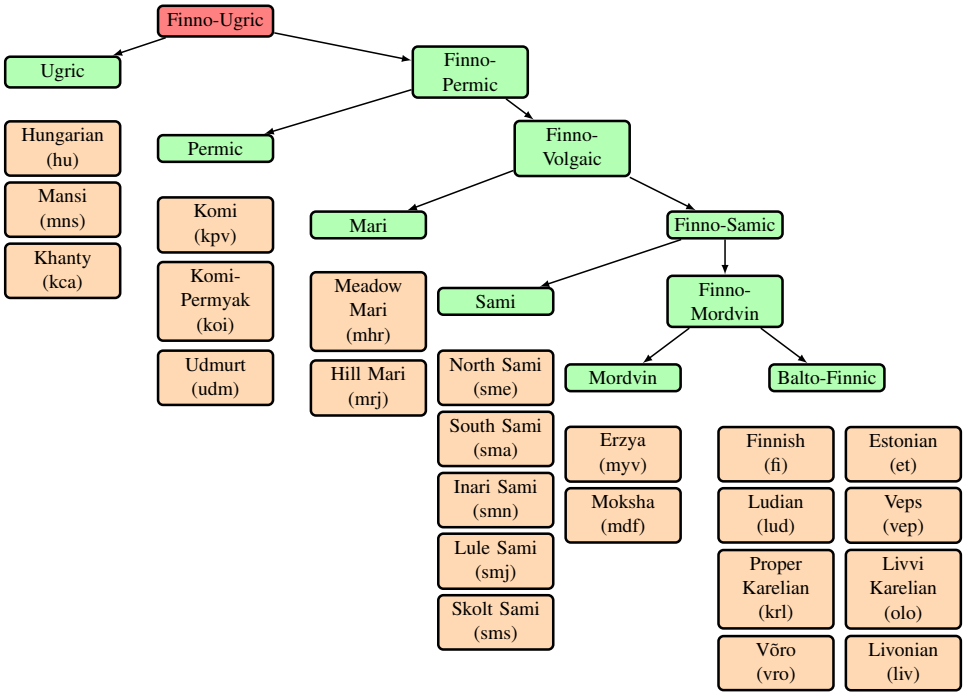


Figure 1: Languages from the Finno-Ugric language family, for which we have created MT systems. Green colour represents branches, orange — languages. The Finno-Permic languages are visualized according to the Janhunen classification (Janhunen, 2009).

3.1 Monolingual corpora

We collected monolingual corpora mainly by crawling texts off the web and combining with pre-existing corpora. Three main categories of texts can be distinguished: news, Wikipedia, and biblical. Texts that do not fall into these categories have been grouped together under the category "Other". Table 1 provides more information of the amount of data collected.

Wikipedia texts were collected from the Wortshatz corpora collection (Goldhahn et al., 2012) and the Tatoeba Translation Challenge corpora (Tiedemann, 2020).

The biblical subcorpus consists of texts taken from the Finugorbib[1] and the open corpus of Veps and Karelian languages "VepKar" (Boyko et al., 2022)[2].

In order to create a subcorpus of news, we used the following online news media:

- **Komi (kpv)**: http://komikerka.ru/, https://komiinform.ru/news/e/161, https://www.nbrkomi.ru/kraevedenie/vyltoryas
- **Udmurt (udm)**: https://udmddn.ru/ivorjos/, https://oshmes.info/
- **Erzya (myv)**: https://vk.com/club78443596
- **Moksha (mdf)**: https://mokshapr.ru/
- **Livvi Karelian (olo)**: https://www.omamedia.ru/ka/
- **Veps (vep)**: https://www.omamedia.ru/ve/
- **Mansi (mns)**: https://khanty-yasang.ru/
- **Khanty (kca)**: https://khanty-yasang.ru

The subcorpus "Other" is a collection of texts from the Mozilla dataset of voices "Common Voice"[3] and the open corpus of Veps and Karelian languages "VepKar".

Monolingual data for most of the high-resource languages (English, Estonian, Finnish, Hungarian, Latvian, Russian) was sampled from the WMT news dataset[4]. The Norwegian monolingual data was sampled from the "Norsk aviskorpus"[5]. Parallel data between high-resource languages was sampled from OPUS (Tiedemann, 2012).

We share the part of the monolingual corpora[6].

[1] http://www.finugorbib.com/alt/alt_al.html
[2] http://dictorpus.krc.karelia.ru/en
[3] https://commonvoice.mozilla.org/en/datasets
[4] https://data.statmt.org/news-crawl/
[5] https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-4/
[6] https://huggingface.co/datasets/tartuNLP/smugri-data

	mono				
	wiki	bible	news	others	total
kpv	18.4	4.5	38.3		61.2
koi	11.5	1.2			12.7
udm	43.5	3.7	36		83.2
mrj	49.5			14.6	64.1
mhr	141			109	251
myv	73.8		1.3	7.7	82.8
mdf	8	3.9	3.9	0.3	16.1
krl		1.8		18.4	20.2
lud				5.3	5.3
olo			21	19.4	40.4
vep	71.3	0.9	7.8	35.3	115.3
vro				162	162
liv				40	40
sma				55	55
sme				34	34
smj				128	128
smn				123	123
sms				76.7	76.7
mns		0.8	10.3		11.1
kca		0.8	13.3		14.1

Table 1: The collected monolingual corpus of the low-resource languages. The figures in the table are in thousands of sentences.

3.2 Parallel Corpora with Russian

As the majority of speakers of the low-resource FU languages live in Russia, most of the parallel translations we have collected are in Russian (see Table 2). A substantial portion of the parallel corpus consists of biblical texts from the Finugorbib and "VepKar". The rest of the parallel corpus comprises various texts, mostly collected from the "VepKar" and Finnougoria webpage[7].

3.3 Data for Võro, Livonian, and Sami Languages

The data (parallel and monolingual) for the Võro, Livonian, and Sami languages that we included in our experiments were taken from the previous editions of NMT developments with low-resource FU languages (Tars et al., 2021, 2022b; Rikters et al., 2022). Võro data is mostly from a META-SHARE[8] source consisting of newspapers, fiction, and a handful of other domains. Livonian data comes from OPUS (Tiedemann, 2012) Liv4ever dataset. Sami language data was collected in previous works from the resources of The Arctic University of Norway[9].

[7]https://finnougoria.ru/

	parallel (Ru)		
	bible	others	total
kpv	11	2	13
koi	8	0.3	8.3
udm	30		30
mrj	8		8
mhr	9		9
myv	11.5	0.9	12.4
mdf	11.5	1	12.5
krl	10.5	7.7	18.2
lud		10.5	10.5
olo	11.9	4	15.9
vep	16.4	11.1	27.5
mns	0.7		0.7
kca	2		2

Table 2: The collected parallel corpus with Russian. The figures in the table are in thousands of sentences.

4 Benchmark dataset

In order to create a multilingual benchmark for Finno-Ugric languages[10], we took the first 250 rows of the FLORES dataset (NLLB Team et al., 2022) and had them translated into nine Finno-Ugric languages: Komi, Udmurt, Hill and Meadow Mari, Erzya, Moksha, Livonian, Mansi, and Livvi Karelian by a team of bilingual speakers, both natives and fluent speakers, of Estonian or Russian and low-resource FU languages. Most translators have an academic degree in linguistics or have extensive translation experience.

While translating, translators have encountered the following problems:

1) Some sentences of the FLORES dataset contain very specific vocabulary, such as "barbs" or "barbules", which can be hard to translate because

[8]https://doi.org/10.15155/1-00-0000-0000-0000-001A0L

[9]https://giellalt.uit.no/tm/TranslationMemory.html

[10]https://huggingface.co/datasets/tartuNLP/smugri-flores-testset

the translators are unfamiliar with this scientific domain.

2) Some words, such as "inning" or "shuttle", are not commonly used or have never been used in some FU languages. As a result, translators have had to create new words based on their sense of the language.

3) The FLORES dataset contains a few lengthy sentences, whereas, in some FU languages, it is preferable to use shorter sentences. So the long sentences have been divided into shorter sentences

While working on creating new benchmark datasets, we found a broken row in the original English dataset: "Singer Sanju Sharma started the evening, followed by Jai Shankar Choudhary. *esented the chhappan bhog bhajan as well*. Singer, Raju Khandelwal was accompanying him." As we can see, the second sentence makes no sense. To fix it, (i) we have omitted this sentence in the English, Latvian, Norwegian (Bokmål), and Russian datasets; (ii) we have added the translation of the last sentence, which was missing, to the Estonian dataset ("Õhtut alustas laulja Sanju Sharma, kellele järgnes Jai Shankar Choudhary. Laulja Raju Khandelwal oli teda saatmas"); (iii) we have edited the first sentence in the Finnish dataset by removing part of it ("Illan aloitti laulaja Sanju Sharma, ja häntä seurasi Jai Shankar Choudhary ~~, joka esitti myös chhappan bhogien bhajanin~~. Häntä säesti laulaja Raju Khandelwal."); (iv) we have replaced the second sentence in the Hungarian dataset ("Sanju Sharma énekes indította az estét, õt követte Jai Shankar Choudhary. ~~pedig a chhappan bhog bhajant adta elő Raju Khandelwal kíséretében.~~ Raju Khandelwal énekes kísérte.").

5 Experiments

One of the goals of our paper was to find out which language pairs are needed to reach a certain level of quality for low-resource NMT models in the Finno-Ugric setting. More specifically, the question is whether it is necessary for low-resource multilingual systems to back-translate in all directions (which is costly) or subsets of translation directions can suffice? By finding optimal amounts of synthetic data we can optimize the overall system creation process by making it less costly and less time-consuming while being able to increase the number of iterations performed.

5.1 Experiment setup

The baseline in this work is a pre-trained multilingual neural machine translation model (M2M-100, 1.2 billion parameters) that has been fine-tuned on parallel data of previously unseen language pairs in addition to sampled high-resource language pairs to reduce catastrophic forgetting (20k samples per high-resource language pair).

For the back-translation experiments, we designed four sets of back-translation data:

1. Synthetic data between all languages (702 language pairs) (`all-all`).
2. 10% of synthetic data of every language pair in the first set (`all-all-10`).
3. Synthetic data from each low-resource language to each high-resource language and vice versa (for example Udmurt-English, Estonian, Finnish, Latvian, Norwegian, Hungarian, Russian) (`L-H`).
4. Synthetic data from each low-resource language to its related high-resource languages and languages it had original parallel data with and vice versa (for example Udmurt-Estonian, Finnish, Russian) (`L-rH`).

All of the sets had an upper limit of 100k synthetic segments per language pair.

The third and fourth sets were chosen a bit more strategically, incorporating linguistic knowledge about the low-resource languages. The third set was created to see whether high-resource monolingual data helps the low-resource languages more efficiently when we do not have other data distracting the model. The fourth set included synthetic data for each low-resource language to its related high-resource languages plus language pairs that it already had parallel data with.

5.2 Technical specifications

We trained all the described NMT systems on the LUMI[11] supercomputer. All models were fine-tuned with the Fairseq framework (Ott et al., 2019) implementation of M2M-100 (Fan et al., 2021) for 350k updates with a batch size of 3840 tokens (the number was chosen to match earlier versions of models trained with the Huggingface implementation of M2M-100). All models were fine-tuned on 4 AMD Mi250X GPU-s. We used custom

[11] https://www.lumi-supercomputer.eu/about-lumi/

scripts[12] to expand the embedding matrix and the vocabulary of M2M-100.

6 Results

Quantitative analysis To get an overview of the quality of the models and compare different synthetic data settings, we compare chrF++ (Popović, 2015, 2017) results for all of the experiments, calculated using sacreBLEU (Post, 2018)[13]. As we are evaluating morphologically rich languages, reporting chrF++ as the main automatic metric gives the most truthful results, whereas BLEU (Papineni et al., 2002) is too punishing on this type of languages.

In Table 3, we display comparisons of all five models (baseline and four models with different synthetic datasets) with different clusterings of language pairs.

In the subtable 3a, we notice that adding synthetic data from every language pair damages the translation quality translating into low-resource languages. Comparing `all-all` and `all-all-10` models, where `all-all-10` contains 90% less synthetic data, higher quality is obtained by the `all-all-10` over all of the language pairs as well as translating into low-resource languages. This means that better results are achieved with less synthetic data and less training time/resources used.

The cause of this situation is the fact that although we limited monolingual data to 100k for each language pair, some smaller language pairs had a lot less than 100k monolingual sentences. Taking only 10% of the synthetic data leveled the distribution of high- and low-resource synthetic data and allowed high-resource to low-resource pairs to get more attention during training.

The best scenario for translating into low-resource languages seems to be to use synthetic data from low-resource language into related high-resource languages (`L-rH`). This is shown by the subtables 3a and 3c. For translating from low-resource languages to high-resource languages, however, the most efficient is to add synthetic data from each low-resource language to all the high-resource languages involved in the initial fine-tuning (`L-H`), instead of using the larger `all-all` model.

[12] https://github.com/TartuNLP/m2m-100-finetune

[13] sacreBLEU signature:
`nrefs:1|case:mixed|eff:yes|nc:6|nw:2|space:no|version:2.0.0`

Comparing the baseline to all the other models, we see significant improvements which can be explained by the fact that the parallel data for low-resource languages originated mainly from the bible, but monolingual data originated from different domains, even more for the high-resource languages.

One anomaly clear from subtable 3c, is the Mansi language performing badly across all of the models with the highest score being 10+ points below the scores for other languages. After further inspection, the fault seemed to be the non-normalized symbols in the dataset which were not included in the dictionary before training and were causing unknown symbols in the translations.

We do not report results for low-resource languages that lack the FLORES benchmark, because the held-out test set is too biased towards the bible domain and there is no other comprehensive benchmark for the rest of the low-resource languages.

In addition to the mentioned experiments, we tried filtering the back-translation data with some of the same filters used to filter the original parallel data. However, the results of the experiments with filtered back-translation data were the same or even a little worse than with the non-filtered back-translation data. Thus, we do not report these results and leave the thorough back-translation filtering analysis for future work.

Comparison to previous results To compare some of the language pairs to previous results on already existing test sets (Tars et al., 2022b,a), we offer a detailed overview of high- to low-resource translation directions for Võro, Livonian, and all the included Sami languages in Table 4. The improvement with our model varies between the language pairs, but the majority of the compared directions achieve a noticeable gain in BLEU, some even very significant 10 and 20 BLEU point increases which might indicate some test data leakage into the training set. The improvements for English-Livonian are noteworthy because although our model gains only about 0.5 BLEU points, it was trained with fewer back-translation iterations and did not need extra finetuning to the specific language pair. For other translation directions, it can be hypothesized that the improved scores are a result of adding synthetic data because the methods we are comparing to omitted using back-translation.

	low-low	low-high	high-low	low-high(rel)	high(rel)-low	all pairs
baseline	18.7	24.0	20.7	24.8	22.1	23.7
all-all	20.2	36.5	19.1	36.8	20.0	28.5
all-all-10	25.9	34.3	24.1	34.9	25.5	30.3
L-H	26.6	**36.6**	25.8	**37.0**	27.2	**32.3**
L-rH	**27.2**	35.5	**26.8**	36.1	**28.2**	32.0

(a) low - low-resource languages, high - high-resource languages, rel - related languages to respective low-resource language. "-" indicates two-way translation directions between the languages.

	to-RU	to-EN	to-ET	to-FI	to-HU	to-LV	to-NO
baseline	19.6	25.6	26.6	25.1	22.0	24.3	24.8
all-all	42.3	39.8	**28.2**	36.8	35.2	37.6	**35.4**
all-all-10	39.2	37.5	27.8	34.7	32.4	35.2	33.6
L-H	**42.9**	**40.4**	27.8	**37.2**	**35.5**	**38.0**	34.6
L-rH	41.8	39.4	26.7	36.4	33.8	35.9	34.6

(b) to-* indicates translation directions from low-resource languages to the respective high-resource language.

	to-KPV	to-LIV	to-MDF	to-MHR	to-MNS	to-MRJ	to-MYV	to-OLO	to-UDM
baseline	15.9	28.4	22.1	21.3	12.2	19.9	22.9	22.7	21.0
all-all	15.9	26.0	18.2	24.4	12.4	15.2	16.7	21.2	21.5
all-all-10	22.3	28.6	25.2	28.3	13.7	22.1	23.1	25.3	28.1
L-H	24.6	29.5	27.0	**30.8**	14.3	23.8	24.4	**27.1**	31.2
L-rH	**26.4**	**29.7**	**28.5**	30.6	**16.1**	**26.2**	**25.2**	26.7	**31.6**

(c) to-* indicates translation directions from high-resource languages to the respective low-resource language.

Table 3: Average chrF++ results for all experiments across different language pair clusters on FLORES benchmarks. **Bold** - highest score per grouping. all-all - contains BT data from every language pair. all-all-10 - contains 10% of BT data used in all-all. L-H - contains BT data from each low-resource language to each high-resource language and vice versa. L-rH - contains BT data from each language to its related high-resource language + high-resource languages it had parallel data with and vice versa.

	en-liv	et-liv	et-vro	fi-sma	fi-sme	fi-smn	fi-sms	no-sma	no-sme	no-smj
L-rH	**15.74**	**24.17**	30.63	**46.58**	38.27	**67.34**	**44.13**	**60.79**	35.21	**51.95**
previous best	15.19	14.51	**34.11**	26.63	**42.89**	53.3	33.72	46.79	**35.38**	40.01

Table 4: BLEU scores for high-resource to selected low-resource languages to compare with previous results in these language pairs. The previous best results are from Tars et al. (2022b,a). The test set is same as used in the previously mentioned publications. **Bold** - best result between our best high-low model and the previous best result.

Qualitative analysis Here, we go over the key findings of the qualitative analysis we conducted. We focus and showcase our results on the Komi to Russian translation direction and compare our baseline model and the model trained on back-translated data. The baseline model performed poorly with unnatural and "biblical-looking" translations which is a style introduced by the parallel training data used for the baseline. The baseline model output sounds like Church Slavic, which is a Slavic liturgical language used by the Eastern Orthodox Church, examples of this are both in Figures 2 and 3. The baseline model also introduces biblical artifacts into the translation, which is showcased by an example shown in Figure 2, where "Daesh" is changed

Original (kpv)	Полиция юӧртіс, мый уськӧдчӧмын найӧ мыжалӧны чайтана боевикӧс Даешысь (ИГИЛ).
Baseline (ru)	Полиция объявила, что его обвиняют в убийстве в Иерусалиме.
BT (ru)	Полиция сообщила, что в нападении они подозревают предполагаемого боевика группировки "Даеш" (ИГИЛ)
Reference (ru)	Полиция заявила, что в совершении нападения подозревается предполагаемый боевик ДАИШ (ИГИЛ).
Reference (en)	Police said they suspect an alleged Daesh (ISIL) militant of responsibility for the attack.

Figure 2: Example of translations from Komi to Russian. The `Baseline` translation is partially correct. We highlight the word "Jerusalem" in red as it is an artifact (hallucination originating from the Bible) created by the model. The `BT` translation is generally correct, with a small error in the word Daesh, which is highlighted in green. BT refers to the back-translation model, specifically the L-rH model.

Original(kpv)	Ӧти воӧн висьысь морт матысса йитчигӧн вермӧ висьмӧдны 10-сянь 15 мортӧдз.
Baseline (ru)	И если гонщик вошел в дом ближнего своего, то есть человек, лежащий в язве или в болезни, то есть около десятого, или около пятнадцатого;
BT (ru)	В течение одного года риск заражения человека при близких контактах может возрастать с 10 до 15 человек.
Reference (ru)	За один год инфицированный человек может заразить от 10 до 15 человек при близком контакте.
Reference (en)	In one year's time, an infected person may infect 10 to 15 close contacts.

Figure 3: Example of translations from Komi to Russian. The translation by the `Baseline` model is generally incorrect, and it is written in the biblical style. The words that stand out as biblical are highlighted in red. The `BT` translation is completely correct. BT refers to the back-translation model, specifically the L-rH model.

into Jerusalem. We found multiple occurrences of Jerusalem in the baseline translations but none of such occurrences in the translations made by the model with the additional back-translated data. Our proposed model, which added a lot of synthetic data into training data, produces much better translations — we hypothesize that this is due to better distribution of data sources, the translations look more general and have an informative style. We also did not notice any named-entity hallucinations. Our findings highlight the importance of data source (domain) and quality in the low-resource scenario, where imbalanced data sources can lead to non-optimal translations.

7 Conclusion

We presented a FLORES-based benchmark dataset for nine low-resource Finno-Ugric languages: Erzya, Komi, Livvi Karelian, Livonian, Hill and Meadow Mari, Mansi, Moksha, and Udmurt. In this study, we trained and evaluated multiple models for these languages and generated a large amount of synthetic parallel data through back-translation. The results showed that the models achieved promising performance on the benchmark dataset and demonstrated the potential of these methods for low-resource machine translation. Our experiments also showed that it could be useful to choose back-translation settings more strategically, selecting certain language pairs, to achieve better results while using fewer resources for back-translation and training.

Limitations

The machine translation systems described in this paper have several limitations that are important to consider.

- Most of the parallel training data comes from the Bible - this limits the generalizability of the system, for example when trying to translate non-religious texts from Wikipedia.
- Train-Test mismatch, specifically for the parallel training data, impacts the overall trustworthiness of the quantitative results.
- Limited test data coming from a single source - We managed to translate only a quarter of the multilingual FLORES dataset. Also, we only have the FLORES dataset which originates from [English] Wikipedia.
- Finno-Ugric languages written in the Cyrillic alphabet might benefit from transliteration, which we did not try in this study. Transliteration converts text written in one script into another script. It remains an open question if transliteration into the Latin script would improve the translation quality.

These limitations highlight the need for further research in machine translation for Finno-Ugric languages. Future studies should address these limitations.

References

Tatyana Boyko, Nina Zaitseva, Natalia Krizhanovskaya, Andrew Krizhanovsky, Irina Novak, Nataliya Pellinen, and Aleksandra Rodionova. 2022. The open corpus of the veps and karelian languages: Overview and applications. *KnE Social Sciences*, 7(3):29–40.

Laurie Burchell, Alexandra Birch, and Kenneth Heafield. 2022. Exploring diversity in back translation for low-resource machine translation. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 67–79, Hybrid. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana. Association for Computational Linguistics.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. 2022. Survey of low-resource machine translation. *Computational Linguistics*, 48(3):673–732.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Juha Janhunen. 2009. Proto-uralic—what, where, and when. *The quasquicentennial of the Finno-Ugrian society*, 258:57–78.

Tom Kocmi, Rachel Bawden, OndÅ™ej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal NovÃ¡k, Martin Popel, Maja PopoviÄ‡, and Mariya Shmatova. 2022. Findings of the 2022 conference on machine translation (wmt22). In *Proceedings of the Seventh Conference on Machine Translation*, pages 1–45, Abu Dhabi. Association for Computational Linguistics.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia-Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Matīss Rikters, Marili Tomingas, Tuuli Tuisk, Valts Ernštreits, and Mark Fishel. 2022. Machine translation for Livonian: Catering to 20 speakers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 508–514, Dublin, Ireland. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Maali Tars, Taido Purason, and Andre Tättar. 2022a. Teaching unseen low-resource languages to large translation models. In *Proceedings of the Seventh Conference on Machine Translation*, pages 375–380, Abu Dhabi. Association for Computational Linguistics.

Maali Tars, Andre Tättar, and Mark Fišel. 2021. Extremely low-resource machine translation for closely related languages. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 41–52, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Maali Tars, Andre Tättar, and Mark Fišel. 2022b. Cross-lingual transfer from large multilingual translation models to unseen under-resourced languages. *Baltic Journal of Modern Computing*, 10.3:435–446.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Jörg Tiedemann. 2020. The Tatoeba Translation Challenge – Realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Rui Wang, Xu Tan, Renqian Luo, Tao Qin, and Tie-Yan Liu. 2021. A survey on low-resource neural machine translation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4636–4643. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Distilling Estonian Text Domains for Production-Oriented Machine Translation

Elizaveta Korotkova **Mark Fishel**
Institute of Computer Science
University of Tartu, Estonia
{elizaveta.korotkova, mark.fisel}@ut.ee

Abstract

This paper explores knowledge distillation for multi-domain neural machine translation (NMT). We focus on the Estonian-English translation direction and experiment with distilling the knowledge of multiple domain-specific teacher models into a single student model that is tiny and efficient. Our experiments use a large parallel dataset of 18 million sentence pairs, consisting of 10 corpora, divided into 6 domain groups based on source similarity, and incorporate forward-translated monolingual data. Results show that tiny student models can cope with multiple domains even in case of large corpora, with different approaches benefiting frequent and low-resource domains.

1 Introduction

The quality of neural machine translation (NMT, Vaswani et al., 2017) systems heavily depends on training data and the text domains covered in it. Large-scale NMT Transformer models are usually trained on multiple corpora representing different domains (Kocmi et al., 2022), which in turn requires training models with higher representation capacity and an exceedingly large number of parameters, sometimes in the tens of billions for the largest models (Fan et al., 2020; NLLB Team et al., 2022).

However, using such models for inference in a production setting becomes more costly and cumbersome with increasing size. In parallel to the challenge of using more representational and learning power, a constraint from the practical side is to have models be as small and fast as possible for efficient deployment in production.

An additional challenge arises from the variability of natural language and different text domains and styles. While methods for training an NMT model to perform well on a particular type of text are relatively straightforward, the requirement of having a single NMT model translate multiple varieties of input text without a significant loss of quality on any of them due to interference between the domains in the training data remains more difficult.

In this paper, we aim to bridge the gap between previous research and systems applicable in production by experimenting with multi-domain knowledge distillation for NMT on the example of Estonian-English translation. We show that even for tiny NMT student-models and large-scale training data, it is efficient to train a single student model on data distilled by multiple fine-tuned domain-specific teacher models.

Our contributions are:

- we experiment with distilling the knowledge of multiple domain-specific teacher models within a single student model, focusing on very small student models;
- we use a sizeable parallel dataset of 18M sentence pairs, consisting of 10 corpora, which we divide into 6 groups based on similarity of their sources;
- we release our student models, test set translations, and generation code[1].

2 Related Work

Knowledge Distillation for Machine Translation Knowledge distillation (Bucila et al., 2006; Hinton et al., 2015) is the technique of compressing the knowledge learned by a large model with high capacity and a large number of parameters or by an ensemble of models into a single smaller model. Knowledge distillation allows for

[1] https://github.com/TartuNLP/multidomain-students

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 772–781

increased speed and efficiency at inference time, while aiming to not sacrifice the quality of model performance to a significant extent.

Knowledge distillation methods were extended to the task of machine translation by Kim and Rush (2016). One of the methods they proposed is interpolated sequence-level knowledge distillation, consisting of several steps:

- a large teacher model is trained on a corpus of parallel texts;
- the teacher model is used to translate the source side of the parallel corpus into the target language;
- a smaller student model is trained using the original data as source and the teacher-generated (distilled) data as target.

In this way, the student model is trained with the goal of imitating the teacher model's probability distribution over the translations, thus constraining the task from the full space of natural language translations to the much smaller space of translations generated by the teacher, and making it more easily achievable for the small student model.

We follow the sequence-level knowledge distillation procedure proposed by Kim and Rush (2016) in our knowledge distillation experiments.

Recent advances in efficient MT have extended the limits of small and fast NMT models (Junczys-Dowmunt et al., 2018; Kim et al., 2019a; Heafield et al., 2021, 2022), using knowledge distillation, increasingly lightweight architectures and CPU optimization for faster inference, while suffering increasingly small quality decrease compared to full-scale models. However, these experiments are typically focused on single-domain or general-domain translation; we build on the findings of MT efficiency research and use them in a multi-domain scenario.

Knowledge Distillation for Multi-Domain Machine Translation While training a neural machine translation model to perform reasonably well on one specific type of text is relatively straightforward, generalizing to multiple domains within a single model is more challenging. Typically, full-scale NMT models are trained on vast amounts of parallel data representing various text domains (Akhbardeh et al., 2021; Kocmi et al., 2022). Numerous methods which aim to improve multi-domain MT performance have been proposed (Kobus et al., 2017; Tars and Fishel, 2018; Britz et al., 2017; Zeng et al., 2018).

The task of achieving good performance on multiple text domains, together with the need for fast and efficient translation, have lead to combining the methods of multi-domain neural machine translation and knowledge distillation.

Wang et al. (2019) focus on the task of multi-domain translation, using knowledge distillation for additional domain supervision: the probabilities (soft targets) produced by domain-specific models are used when training the unified multi-domain model.

Gordon and Duh (2020) adapt student models to one text domain at a time. They suggest distilling general-domain data to improve the performance of the general-domain student model, fine-tuning the best obtained model to in-domain data, and fine-tuning the teacher model to a specific domain and distilling this in-domain model. In our experiments, we follow Gordon and Duh in fine-tuning teacher models to domain-specific corpora and distilling them.

Our work shares the most similarities with Currey et al. (2020). Similarly to them, we fine-tune a general teacher to obtain several domain-specific teachers, which we then distill into a single student model. However, our work is closer to a real-world production scenario: we use a significantly larger training corpus which combines more individual parallel corpora from different sources, as well as much smaller student models, whereas Currey et al. train teacher models with 12 encoder and 12 decoder layers (roughly 100M model parameters), and their student models follow the Transformer-base configuration (6 encoder and 6 decoder layers, around 60M parameters).

Concurrently to Currey et al. (2020), Mghabbar and Ratnamogan (2020) also explore distilling several domain-specific teachers into a single student model, but use word-level instead of sentence-level knowledge distillation, and do not focus on decreasing the size of student models.

3 Methods

In our experiments, we aim to create neural machine translation models which 1) perform well on several data domains, and 2) are small and efficient.

To distill NMT models, we follow the sequence-

level knowledge distillation framework initially proposed by Kim and Rush (2016), where a smaller student model is trained using the synthetic target-side data produced by a larger teacher model, and experiment with distilling multiple domain-specific teacher models into a single student model.

We follow Currey et al. (2020) in employing a straightforward strategy:

1. train a general-domain teacher model,
2. fine-tune the teacher model to partitions of the data to obtain multiple domain-specific teacher models,
3. use the domain-specific teachers to forward-translate the data,
4. distill the domain-specific teachers into a single student model.

However, our student models are much smaller than the student models used by Currey et al. (2020), and we use significantly more data, bringing our setup closer to a full-scale real-world scenario.

3.1 Data

The experiments are performed on the Estonian-English language pair. We use 10 parallel corpora: Europarl (Koehn, 2005), JRC-Acquis (Steinberger et al., 2006), OpenSubtitles (Lison and Tiedemann, 2016), ParaCrawl (Esplà et al., 2019), EMEA, DGT, infopankki, GNOME, KDE4, and Ubuntu (Tiedemann, 2012). We divide the corpora into 6 groups as shown in Table 1. Europarl forms its own group of parliament proceedings texts (EU), EMEA a group of medical texts (MED), and OpenSubtitles a group of film and TV subtitles (SUBS). We merge the DGT and JRC-Acquis corpora into a group representing legal texts (LEGAL), ParaCrawl and infopankki represent texts crawled from the web (WEB), and, finally, GNOME, KDE4, and Ubuntu form the group of software localization texts (IT).

Table 1 also shows the number of sentence pairs in each group and corpus after cleaning; the total size of the parallel training corpus is ∼18M sentence pairs. The resulting corpus is highly unbalanced, with sizes of the groups varying from ∼100K examples for IT to ∼7.5M for WEB, which is realistic in a production scenario. From each corpus, we separate a development set of 1000 sentence pairs and a test set of 500 sentence pairs. In addition to the held-out development sets, we also include the development split of WMT18 ET-EN set in the validation set. The test part of WMT18 ET-EN is used as an external test set.

3.2 Models

To train our models, we use the Marian framework (Junczys-Dowmunt et al., 2018). First, we train a teacher model from scratch using the 18M training data described above (this model is denoted as T-18M in Table 2). The teacher is a Transformer model, with shared SentencePiece (Kudo and Richardson, 2018) vocabulary of size 32,000 units, 6 encoder and 6 decoder layers, embedding dimension 512, feed-forward dimension 2048, 8 attention heads. The training is stopped when either BLEU (Papineni et al., 2002) or the mean word cross-entropy score on the validation set has not improved for 10 checkpoints, and the best checkpoint is chosen based on validation BLEU.

We then fine-tune the obtained teacher to each of the six data groups (the resulting domain-specific teachers are denoted, for example, T-EU or T-SUBS in Table 2). Fine-tuning is stopped when the validation metrics have not improved for 15 checkpoints. Next, we follow the interpolated sequence-level knowledge-distillation procedure (Kim and Rush, 2016): we forward-translate the parallel training data with the original general-domain teacher or with the corresponding fine-tuned teachers, generating 8-best lists for each source example. The best translation for each sentence is chosen based on its similarity to the original target sentence according to sentence-level BLEU.

In addition to the parallel data described above, we forward-translate 1M Estonian sentence pairs from the News Crawl corpus (articles from 2019 and 2020) and add those to the data the student models are trained on. (In this case, we cannot choose the translations which are closest to the original target, as no original target exists.) We also try fine-tuning the teacher model to these news data, where the target side was generated by the teacher itself.

Finally, we train several student models using the original source data and synthetic forward-translations obtained using the teacher models. For efficiency purposes, we follow Kim et al.

group/corpus	corpus size	group size	domain
EU			
Europarl	593,637	593,637	parliament proceedings
LEGAL			
DGT	2,241,448	2,637,222	legislation
JRC-Acquis	395,774		
MED			
EMEA	211,722	211,722	pharmaceutical documents
SUBS			
OpenSubtitles	6,868,517	6,868,517	film & television subtitles
WEB			
ParaCrawl	7,601,013	7,614,325	Web-crawled texts
infopankki	13,312		
IT			
GNOME	3,036		
KDE4	99,808	105,906	software localizations
Ubuntu	3,062		
Total		18,031,329	

Table 1: Sizes of corpora and corpus groups (number of sentence pairs) used for training the ET-EN teacher and student models, after cleaning

(2019b) and replace the self-attention mechanism in the Transformer encoders, which have 6 layers, with GRU-based cells, and use Simpler Simple Recurrent Units in the transformer decoders, which consist of 2 layers. The training is stopped if the metrics have not improved for 20 checkpoints. The resulting student models have disk size of 65 megabytes. S0 is the model trained using data produced by the initial teacher. S-FT uses the data forward-translated by the corresponding fine-tuned teacher for each of the corpora. S-FT-bal uses the same data, but after balancing the corpora: the total size of the training data is kept approximately the same as before, but each data group is downsampled or upsampled so that the sizes of all groups are approximately equal. The last model, S-ORIG, is trained for comparison on original (not forward-translated) parallel data.

We provide our S0, S-FT, and S-FT-bal student models, test set translations generated by them, and code used to generate and evaluate those translations at `https://github.com/TartuNLP/multidomain-students`.

4 Results

Table 2 shows the BLEU scores (Papineni et al., 2002) our teacher and student models achieve on the held-out and WMT18 test sets[2]. We can observe that fine-tuning on a data group noticeably improves the performance of the teacher model on held-out test sets within that data group. Not unexpectedly, the effect is more pronounced for smaller corpora, which are less represented in the whole corpus on which the original mixed-domain teacher is trained. We assume that the second important factor is the extent to which the corpus is narrowly specialized. For example, on the test set extracted from the very small and highly specific Ubuntu corpus, the performance of the fine-tuned teacher model is higher than that of the general teacher by a huge margin of 17.9 BLEU points, while the same performance gap is 3.1 points for OpenSubtitles and 1.9 points for Europarl.

It seems that fine-tuning the teacher on forward-translated monolingual data yields no positive effect. BLEU score on the WMT18 test set remains the same as for the general teacher, while scores on the held-out test set drop. This is not unexpected, as, while the teacher stops encountering data from other domains during fine-tuning, it also only encounters the data from the news domain that it forward-translated itself, and most likely cannot learn to exhibit any new behaviour on these data.

[2]sacreBLEU signature: `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1`

group	EU	IT			LEGAL		MED	SUBS	WEB		NEWS	
corpus	Europarl	GNOME	KDE4	Ubuntu	DGT	JRC-Acquis	EMEA	OpenSubtitles	ParaCrawl	infopankki	WMT18	avg
T-18M	40.9	33.8	29.7	37.8	44.2	57.5	46.8	31.9	50.5	30.9	**30.4**	39.491
T-EU	**42.8**	11.0	11.7	13.8	27.6	36.9	17.6	18.4	23.9	20.3	22.6	
T-IT	9.2	**61.2**	**40.6**	**55.7**	6.9	6.5	10.4	10.5	14.3	9.2	9.0	
T-LEGAL	29.4	17.1	12.7	18.3	**49.7**	**64.6**	24.3	8.1	23.8	16.9	16.5	
T-MED	9.6	9.4	7.3	8.4	12.7	15.4	**66.3**	4.4	10.7	7.3	7.2	**48.727**
T-SUBS	22.2	14.6	10.4	15.3	10.8	10.4	12.3	**35.0**	21.7	16.6	24.7	
T-WEB	38.2	29.5	23.5	33.0	36.4	50.2	39.4	24.9	**52.3**	**37.4**	**30.8**	
T-NEWS	37.7	24.3	20.9	27.1	28.8	38.4	29.5	28.7	35.4	27.0	30.4	
S0	**38.4**	29.3	**25.2**	31.4	40.4	54.3	42.6	**29.7**	47	28.8	28.3	35.945
S-FT	38.3	29	25	31.6	**41.3***	**54.5**	41.3	**29.7**	**48.6***	**30.6***	**28.5**	36.218
S-FT-bal	38.2	**45.6***	23.7	**45.3***	38.8†	50.9†	**49.3***	27.3†	41.6†	25†	26.2†	**37.445**
S-ORIG	35.2	28.1	24.1	29.9	38.3	51.2	40.6	29.1	44.5	29.6	24.2	34.073

Table 2: BLEU scores of teacher and student models trained on 18M ET-EN sentence pairs as measured on different test sets. The columns represent the groups and corpora to which the test sets belong, and the rows indicate models. T-18M denotes the initial mixed-domain teacher model. T-EU, T-IT, etc. are teacher models fine-tuned on the corresponding groups of datasets. S0 is a distilled student model trained on texts forward-translated by T-18M. S-FT is a student model trained on data produced by the fine-tuned, domain-specific teacher models. S-FT-bal is trained on the same data as S-FT, but each data group is upsampled or downsampled so that all groups are of equal size, while the total number of training examples stays the same. S-ORIG is a model of the same configuration as the student models, but trained on original (not forward-translated) texts for comparison as a sanity check. The "avg" column shows each model's BLEU, averaged over all test sets (for fine-tuned teachers, we report a single average over the scores of each teacher's translations of test sets belonging to the corresponding group). Bold numbers indicate the highest BLEU scores for each test set among the teacher and among the student models. Individual test set results that show statistically significant improvements ($p \leq 0.05$) of S-FT and S-FT-bal in comparison to S0 are marked with *, while results that are significantly lower than S0 are marked with †.

While the behaviour of fine-tuned teachers is rather straightforward, the performance of student models is more varied. Comparing S0 and S-FT, we observe relatively similar performance: the difference on various test sets ranges from none (OpenSubtitles) to 1.8 (infopankki) BLEU points. The BLEU score averaged over all test sets is better for S-FT, but not by a very large margin. On the external WMT18 test set, S-FT performs best, although it only outperforms S0 by 0.2 BLEU points. On 6 test sets out of 11, S-FT is better than S0, although only on 3 of those the difference is statistically significant (Koehn, 2004), and on one more test set (OpenSubtitles) their result is the same.

The extremely small GNOME and Ubuntu corpora obviously benefit from balancing the data, and the scores on their test sets improve significantly compared to the unbalanced S-FT. Performance on the EMEA corpus, which comprises the second smallest data group, also noticeably benefits from upsampling. At the same time, if we compare the results obtained by S-FT and S-FT-bal on other corpora, we notice drops of 0.1-7.0 BLEU points.

The best average BLEU score is achieved by S-FT-bal, the student trained on data which is forward-translated by the fine-tuned teacher models and balanced.

5 Qualitative Analysis

Table 3 shows several example sentences from test sets belonging to each of the data groups, as well as their reference translations, translations generated by S0, S-FT, and S-FT-bal student models, and sentence-level chrF scores (Popović, 2015). In this section, we provide a brief description of varying model behavior on these examples.

In example 1, which comes from the Ubuntu corpus, only the model trained on balanced data manages to translate ”ruutu soldat” as ”jack of diamonds”, while both S0 and S-FT translate ”ruutu” literally (”squares”), and S0 translates ”soldat” incorrectly (”solder” instead of the direct translation ”soldier”, which is likely due to subword interaction).

In example 2, the S-FT-bal model shows signs of overfitting: the content part of the sentence is identical to the reference, and the number ”63” is generated at the start of the text, where the reference sentence has ”53”. However, there is no number in the source sentence. Sentences produced by the S0 and S-FT models are, in fact, more exact translations of the Estonian source sentence (”you have the feeling”/”you feel” vs. ”you think” and ”the effect of Vivanza”/”Vivanza’s effect” vs. ”Vivanza”).

In the infopankki example (3), all models manage to convey the original meaning of the source sentence, but S-FT-bal does so in a more informal style and with simpler grammar that the reference and the translations by S0 and S-FT (e.g. ”work and business office” vs. ”Employment and Economic Development Office”, and ”helps” vs. ”Help is available”/”You can get help”).

In the example from the OpenSubtitles corpus (4), all models use ”his” instead of ”her” (the Estonian pronoun ”ta” does not have grammatical gender, so the correct English pronoun can only be inferred from wider context). The S-FT model uses the more informal contraction ”it’s”, which is appropriate for the domain. The S-FT-bal model fails to translate a part of the compound word ”kõnepost” and generates ”voice” instead of the correct ”voicemail”.

In example 5 (Europarl), the S-FT-bal model is the only one not to use contractions (”We have” vs. ”We’ve”), which do not typically occur in the formal style of parliament proceedings. However, all three model hypotheses are faithful.

The DGT example (6) sees the S-FT model translate very similarly to the reference, while both S0 and S-FT-bal overgenerate repetitively (”and ovens and ovens” and ”Non-electric non-electric non-electric”).

Finally, in the WMT18 example, all models fail to use the specific correct word ”minesweeper”, and instead translate the compound word ”miinijahtija” more literally as ”mine hunter”. Otherwise, the S-FT hypothesis is the only one to convey the full meaning of the source correctly.

6 Discussion

We observe that the fine-tuned teacher models predictably suffer from forgetting the general teacher’s knowledge on domains other than the one the particular teacher is fine-tuned to. The extent of this forgetting varies, e.g. the teacher fine-tuned to Web-crawled text performs 2.7 BLEU points worse than the mixed-domain teacher on the Europarl test set, while the teacher fine-tuned to medical documents is 27.5 points worse on the

	corpus	model	sentence	chrF
1	Ubuntu	SRC	ruutu soldat	
		REF	jack of diamonds	
		S0	squares solder	4.8
		S-FT	squares of the jack	18.4
		S-FT-bal	the jack of diamonds	95.0
2	EMEA	SRC	Kui teil on tunne, et Vivanza toime on liiga tugev või liiga nõrk, pidage nõu oma arstiga.	
		REF	53 Tell the doctor if you think Vivanza is too strong or too weak.	
		S0	If you have the feeling that the effect of Vivanza is too strong or too weak, talk to your doctor.	58.2
		S-FT	If you feel that Vivanza's effect is too strong or too weak, talk to your doctor.	54.7
		S-FT-bal	63 Tell the doctor if you think Vivanza is too strong or too weak.	98.0
3	infopankki	SRC	Töö otsimisel saab abi Töö- ja ettevõtlusbüroost.	
		REF	The Employment and Economic Development Office provides help with your job hunting.	
		S0	Help is available in the Employment and Economic Development Office.	60.6
		S-FT	You can get help in finding a job at the Employment and Economic Development Office.	62.2
		S-FT-bal	The work and business office helps to seek the job.	16.1
4	OpenSubtitles	SRC	See on ta kõnepost.	
		REF	It's her voicemail.	
		S0	This is his voice mail.	52.6
		S-FT	It's his voice mail.	67.1
		S-FT-bal	This is his voice.	21.6
5	Europarl	SRC	Oleme palju ära teinud, kuid töö ei ole veel läbi.	
		REF	We have come a very long way, but the work is not yet complete.	
		S0	We've done a lot, but the work is not over.	39.0
		S-FT	We've done a lot, but the job's not over.	22.7
		S-FT-bal	We have done a lot, but the work is not over.	44.4
6	DGT	SRC	Mitte-elektriliste töötus- ja laboriahjude ja -põletuskambrite osad	
		REF	Parts for non-electric industrial or laboratory furnaces and ovens	
		S0	Parts of non-electrical furnaces and ovens and ovens	51.3
		S-FT	Parts of non-electric industrial or laboratory furnaces and ovens	90.9
		S-FT-bal	Non-electric non-electric non-electric furnaces and oven parts	46.8
7	WMT18	SRC	Sel poolaastal kuulub rahvusvahelise üksuse koosseisu ka Eesti mereväe miinijahtija Sakala.	
		REF	This half-year, the Estonian minesweeper Sakala is also part of the international unit.	
		S0	This half-year is also part of the international unit Sakala, a mine hunter of the Estonian Navy.	34.3
		S-FT	This half-year the international unit also includes the Estonian naval mine hunter Sakala.	73.7
		S-FT-bal	In this half, the Estonian Navy mine hunter also includes the Estonian Navy mine hunter.	63.7

Table 3: Examples of source-reference pairs from different test sets and corresponding translations produced by S0 (student model trained on texts forward-translated by a single mixed-domain teacher model), S-FT (student model trained on data translated by multiple fine-tuned teacher models), and S-FT-bal (trained on balanced data produced by multiple fine-tuned teachers) models. The last column shows sentence-level chrF score for each of the translations (sacreBLEU signature: `chrF2|nrefs:1|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.3.1`).

OpenSubtitles test set than the original teacher, its score dropping to 4.4 BLEU, which suggests that these translations are not too far from random. The different levels of forgetting could potentially serve as a clue to domain similarity and guide the choice of manual data groupings.

There is a sizeable gap between the average performance of teachers fine-tuned to each data group and the best student model average. While the difference in capacity becomes even more drastic when we compare not one, but several large fine-tuned models to a single small student model, this gap remains large, and suggests the possibility of pushing the limits of student models' performance further.

Distilling the data clearly benefits the training of small models, the S-ORIG model lagging behind other student models. The best average BLEU is achieved by the student model trained on data distilled by multiple fine-tuned teachers and balanced between groups. However, in a real-world scenario trade-offs may still need to be made between the performance on specific domains, with average BLEU score not being representative enough for fine-grained evaluation.

7 Future Work

In our experiments, we used 10 corpora and, for fine-tuning, split them manually into 6 data groups based on the assumed similarity of their sources and topics. However, as demonstrated by Currey et al. (2020), the known domain labels may be suboptimal, and assigning the domains automatically can improve the multi-domain MT performance. Generating automatic domain labels using the general-domain model's internal data representations has been shown to further improve in-domain translation quality (Del et al., 2021). In future work, we would like to explore these methods for automatic domain discovery in conjunction with multi-domain knowledge distillation.

Aiming to bring our experiments closer to a production scenario, we tried incorporating forward-translated monolingual data into our multi-domain distillation setup. However, large-scale systems typically use back-translation and round-trip translation to increase the amount of training data. It currently remains unclear how to best incorporate monolingual data into the multi-domain knowledge distillation framework effectively, given the suboptimal results we achieved when fine-tuning a mixed-domain teacher model to a forward-translated news corpus. We hypothesize that the teacher model cannot learn to exhibit any new behaviours when it is fine-tuned on data generated by itself. Thus, adding monolingual domains to distilled multi-domain systems is a potential topic for future exploration.

Another important direction for future work is extending our research to other language pairs and translation directions. While we perform our experiments on the Estonian→English language pair, which, to the best of our knowledge, has not been experimented with in a similar setting before, using other languages, especially low-resource ones, might lead to different results and insights.

8 Conclusion

In this work, we explored distilling multiple domain-specific neural machine translation teacher models into a single student model. While following procedures proposed in previous work, we incorporated research findings on model efficiency and focused on obtaining very lightweight student models. We used a training corpus of 18M Estonian-English sentence pairs, comprised of 10 unbalanced domains. We separated the domains into groups based on their perceived similarity, explored the effects of balancing, and incorporated monolingual forward-translated data into training of multi-domain students.

Our experiments show that the knowledge of several fine-tuned teachers models can be distilled into a very small student model, with balanced representation of domains further improving the average result. The massive total capacity of several fine-tuned teacher models has a huge average gain over the untuned teacher (almost 10 BLEU points) and the student models with their limited capacity achieve a much more modest increase in translation quality. Still, the increase in translation quality compared to the baseline student is stable and noticeable (+1.5 BLEU points).

Acknowledgements

This work has been supported by the grant No. 825303 (Bergamot[3]) of European Union's Horizon 2020 research and innovation program. The authors thank the Unversity of Tartu's High-Performance Computing Center for providing

[3] https://browser.mt/

GPU computing resources (University of Tartu, 2018). We also thank the anonymous reviewers for their comments and suggestions.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark. Association for Computational Linguistics.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Knowledge Discovery and Data Mining*.

Anna Currey, Prashant Mathur, and Georgiana Dinu. 2020. Distilling multiple domains for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4500–4511, Online. Association for Computational Linguistics.

Maksym Del, Elizaveta Korotkova, and Mark Fishel. 2021. Translation transformers rediscover inherent data domains. In *Proceedings of the Sixth Conference on Machine Translation*, pages 599–613, Online. Association for Computational Linguistics.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland. European Association for Machine Translation.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Çelebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22:107:1–107:48.

Mitchell Gordon and Kevin Duh. 2020. Distill, adapt, distill: Training small, in-domain models for neural machine translation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 110–118, Online. Association for Computational Linguistics.

Kenneth Heafield, Biao Zhang, Graeme Nail, Jelmer Van Der Linde, and Nikolay Bogoychev. 2022. Findings of the WMT 2022 shared task on efficient translation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 100–108, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Kenneth Heafield, Qianqian Zhu, and Roman Grundkiewicz. 2021. Findings of the WMT 2021 shared task on efficient translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 639–651, Online. Association for Computational Linguistics.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019a. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019b. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference*

Recent Advances in Natural Language Processing, RANLP 2017, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl : A Parallel Corpus for Statistical Machine Translation. *MT Summit*, 11.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France.

Idriss Mghabbar and Pirashanth Ratnamogan. 2020. Building a multi-domain neural machine translation model using knowledge distillation. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2116–2123, Santiago de Compostela, Spain. IOS Press.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia-Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006*.

Sander Tars and Mark Fishel. 2018. Multi-domain neural machine translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 259–268, Alacant, Spain.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

University of Tartu. 2018. Ut rocket cluster, https://doi.org/10.23673/ph6n-0144.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yong Wang, Longyue Wang, Shuming Shi, Victor O. K. Li, and Zhaopeng Tu. 2019. Go from the general to the particular: Multi-domain translation with domain transformation networks. In *AAAI Conference on Artificial Intelligence*.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Brussels, Belgium. Association for Computational Linguistics.

Spelling Correction for Estonian Learner Language

Kais Allkivi-Metsoja
School of Digital Technologies
Tallinn University
kais@tlu.ee

Jaagup Kippar
School of Digital Technologies
Tallinn University
jaagup@tlu.ee

Abstract

Second and foreign language (L2) learners tend to make specific spelling errors compared to native speakers. Language-independent spell-checking algorithms that rely on n-gram models can offer a simple solution for improving learner error detection and correction due to context-sensitivity. As the open-source speller previously available for Estonian is rule-based, our aim was to evaluate the performance of bi- and trigram-based statistical spelling correctors on an error-tagged set of A2–C1-level texts written by L2 learners of Estonian. The newly trained spell-checking models were compared to existing correction tools (open-source and commercial). Then, the best-performing Jamspell corrector was trained on various datasets to analyse their effect on the correction results.

1 Introduction

It has been proposed that tailor-made spelling error correction systems are best suited for language learning purposes because the spell-checking tools developed for proficient users often prove unable to correct specific mistakes, like real-word errors, i.e., errors that result in a valid homonym; diacritic errors; or pronunciation-induced errors possibly with a large edit distance (e.g., Lawley 2016). Whereas it is costly to develop rule-based error correction systems with learner-oriented explanations, or neural spell-checking systems that require vast quantities of training data comprising authentic or synthetic errors, statistical spelling correction algorithms which use n-gram language models to analyse context could form a simple starting point for improving error detection and correction of L2 learner writings. In this language-independent approach, only a corpus of (presumably) correct language use samples is needed to train the system.

Currently, the only open-source spell-checker developed for Estonian language is Vabamorf[1]. It is a lexicon- and rule-based library created by Filosoft Ltd. at the beginning of the 1990-s alongside a commercial speller distributed in Microsoft Word (Kaalep et al., 2022). The spellers make use of a lexicon and a list of typing misspellings to assess candidate corrections but they do not appear to rely on context in their suggestions.

For evaluating statistical spelling error detection and correction on Estonian learner language, we first used Peter Norvig's approach that generates all possible spelling corrections by different edits, such as character deletions, insertions, replacements, and transpositions (Norvig, 2007). The procedure is repeated to get correction candidates with two edits. The probability of candidates is estimated based on a unigram language model derived from a language corpus. We used a bigram language model in addition to a unigram model to add context-sensitivity.

Second, we applied the compound aware version of Symmetric Delete Spelling Correction (Symspell)[2]. The algorithm searches for candidate corrections with an edit distance of 1 or 2 based on deletions only, increasing the speed of spelling correction. A corpus-based bigram dictionary can be used, however, bigrams are only considered in ranking suggestions if no suggestions with an edit distance of 1 are found for a single token. Real-word spelling errors are currently not corrected by Symspell (Garbe, 2017).

Third, we tested Jamspell[3] that additionally uses a trigram language model for selecting the highest-scored correction candidate. Jamspell is

[1] https://github.com/Filosoft/vabamorf
[2] https://github.com/wolfgarbe/SymSpell
[3] https://github.com/bakwc/JamSpell

Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 782–788

based on a modified Symspell algorithm, optimized for speed and memory usage, so that the spell-checking library can process up to 5,000 words per second.

We compared the algorithms with three existing spell-checking tools: Vabamorf, and the commercial spellers offered by MS Word (Microsoft 365) and the Google Docs application. The latter uses neural machine translation (Kumar and Tong, 2019).

2 Test Data and Evaluation

The correction output was evaluated on a set of 84 error-annotated proficiency examination writings from the Estonian Interlanguage Corpus[4]. Divided between four proficiency levels (A2, B1, B2, and C1), the texts contain 1,054 sentences, 9,186 words (excluding anonymized identifiers), and 309 spelling errors in total. We distinguished simple spelling errors and mixed errors, i.e., spelling mistakes co-occurring with another error such as word choice, inflectional form, or capitalization error. The error distribution is given in table 1. While the proficiency level increases, the relative frequency of words containing a spelling error decreases, from 5.5% at A2 and 3.8% at B1 to 2.6% at B2 and 2.3% at C1.

Proficiency level	Words	Simple spelling errors	Mixed errors
A2	1,852	73	28
B1	2,186	71	12
B2	2,074	51	3
C1	3,074	68	3
Total	9,186	263	46

Table 1: Spelling correction test data.

The texts have been morphologically annotated in the CoNLL-U format[5], using the Stanza toolkit[6], and manually error-tagged, indicating the error type, scope, and correction in the field for miscellaneous token attributes. While the custom tagset denotes various orthographic and grammatical errors, we only rated the detection and correction of words annotated to have a spelling error (although, we did not count a system edit as unnecessary if the word had any error tag). Each text has been reviewed by two annotators, consulting a third Estonian language expert in case of disagreement. The annotation format allows for several corrections per token but is limited to one error annotation per sentence. This, however, has no significant effect on the analysis of spelling errors, which occur regardless of the sentence structure.

Error detection is the first step of error correction. Nevertheless, to achieve high performance in error detection, the proposed edits do not have to match the gold standard annotation, as opposed to measuring error correction performance. We evaluated both spelling error detection and correction based on three metrics:

- **recall** – the percent of spelling errors detected/corrected;
- **precision** – the percent of relevant/correct changes made;
- **F0.5 score** – a combined measure of precision and recall that gives precision twice as much weight as recall.

The F0.5 score was preferred to the harmonic mean (F1 score) due to the assumption that an error correction system's reliability is rather reduced by false and needless corrections than unproposed corrections (see Ng et al. 2014).

We verticalized the system output and automatically compared it to the test set to detect changes and correction matches. Since L2 learners may not select the correct option from a list of suggestions (e.g., Heift and Rimrott 2008) and such selection cannot be implemented in an automated workflow, e.g., using spell-checking as a pre-processing step of grammatical error correction, we prioritized the speller's accuracy of defining the best correction. Thus, we focused on the highest-ranked suggestion. The cases of mixed errors were reviewed manually to find partial corrections fixing only the spelling of an otherwise erroneous word (e.g., **parnu*∼**pärnu* instead of *Pärnu*, which is an Estonian town name and should be capitalized). Both full and partial word corrections were considered in calculating the evaluation metrics.

3 Comparison of Spell-Checking Tools

The training material for building new statistical spell-checking models came from the Estonian National Corpus (ENC) 2019, which includes web

[4] https://evkk.tlu.ee/about/us/

[5] https://universaldependencies.org/format.html/

[6] https://stanfordnlp.github.io/stanza/

corpora downloaded from Estonian websites as well as the Estonian Reference Corpus, Wikipedia corpora and the corpus of Estonian Open Access Journals (DOAJ) (Koppel and Kallas, 2020). Jamspell and Norvig's spelling corrector were trained on a random sample of 6 million sentences and over 82 million words retrieved from the Reference Corpus that represents the "standard" varieties of Estonian – mostly newspaper texts but also fiction, science and legislation texts from 1990–2008. The sample constitutes nearly half of the Reference Corpus; increasing the volume of the training set did not improve the correction results. Symspell, on the other hand, reached the best results with a uni- and bigram frequency dictionary based on the full ENC 2019 containing over 1.5 billion words. Even then, it performed poorly compared to other tools, especially in terms of recall.

The comparison of spelling error detection and correction by the different applications is summarized in tables 2 and 3. Table 4 shows the distribution of system edits which can be relevant, resulting in identified errors, or unnecessary, leading to broken words. Relevant edits that do not match the expert correction are considered false corrections.[7]

Jamspell and Norvig's speller outperformed Vabamorf and Word's speller in error correction, and Google's spell-checker in error detection. All in all, Google corrected the highest proportion of spelling errors, followed by Jamspell, which still had a significantly better correction recall than the rest of the tools and came close to Google in terms of correction precision and F0.5 score. Despite a larger number of accurate corrections, Google made more than twice as many unnecessary edits.

Spell-checker	F0.5	Precision	Recall
Jamspell	83.9	89.6	67.0
Norvig	78.9	84.3	62.8
Symspell	69.1	86.2	38.5
Google	76.7	78.8	69.6
MS Word	83.4	87.8	69.6
Vabamorf	84.3	89.2	69.3

Table 2: Spelling error detection metrics (%).

[7]The correction outputs as well as the test material can be found at `https://github.com/tlu-dt-nlp/spell-testing/`.

Spell-checker	F0.5	Precision	Recall
Jamspell	64.1	68.4	51.1
Norvig	54.1	57.8	43.0
Symspell	31.4	39.1	17.5
Google	67.5	69.2	61.2
MS Word	51.2	53.9	42.7
Vabamorf	42.6	45.0	35.0

Table 3: Spelling error correction metrics (%).

In error detection, Jamspell yielded results similar to Vabamorf and MS Word. Norvig's spell-checker and Symspell also scored better than Google in detection precision. While Symspell broke the smallest number of words at the cost of very low recall, the lowest percent of unnecessary edits was achieved by Jamspell and Vabamorf – 10.4% and 10.8% respectively. At the same time, 21.2% of words edited by Google did not need to be corrected.

If matching candidate suggestions were considered, the spell-checking tools would reach a higher correction precision, except for Google's speller that proposed only a single correction. Vabamorf's precision (72.5%) would increase the most, Jamspell's precision (72.3%) the least. It means that Jamspell is more likely to suggest an accurate correction with the highest confidence.

Compared to their open-source counterpart Vabamorf, both Jamspell and Norvig's speller benefit from relying on the context of erroneous words. For example, Vabamorf corrected the verbs **tõdida∼tõdeda* 'admit-INF' and **ludeda∼lugeda* 'read-INF' to *tüdida* 'get.bored-INF' and *kudeda* 'spawn-INF'. Interestingly, the rule-based spell-checker tended to replace other parts-of-speech with nouns, e.g., the adverb **lahtii∼lahti* 'open' was changed to *Lahti*, a location in Finland, and the adverb **nanuke∼natuke* 'a bit' to *januke* 'thirst-DIM'. Real-word spelling errors inducing homonymy were best handled by Jamspell that was able to make corrections such as **vaga∼väga* 'very' (*vaga* could be an adjective meaning 'pious, godly'); **töökohtu∼töökohti* 'job-PART.PL' (*töökohtu* could mean 'labour.court-GEN.PL'); and **kuued∼kuud* 'month-PART.SG' (*kuued* could be a numeral meaning 'six-NOM.PL' or a noun meaning 'coat-NOM.PL').

Like Google's spell-checker, Jamspell and Norvig's speller occasionally attempted to cor-

Spell checker	Errors detected	Full corrections	Partial corrections	Broken words
Jamspell	207	129	29	24
Norvig	194	108	25	36
Symspell	119	45	9	19
Google	215	163	26	58
MS Word	215	108	24	30
Vabamorf	214	88	20	26

Table 4: Changes made by spell-checkers.

rect word choice and inflectional form, although merely a couple of mixed errors were fully corrected (e.g., **seles∼sellel laupäeval* 'this Saturday' where the misspelled inessive pronoun was replaced with the correctly spelled adessive form agreeing with the noun). Otherwise, we only took such edits into account if they were unnecessary and resulted in a broken word. It can, however, be noted that Jamspell was more probable to make accurate lexical and grammatical corrections than Norvig's corrector, given a small edit distance, e.g.,**ennem* 'rather'∼*enne* 'before', **kümne* 'ten-GEN.SG'∼*kümme* 'ten.NOM.SG'. Similarly to Vabamorf and MS Word, Norvig's speller replaced some proper nouns with common nouns, e.g., *Kemeris* 'Kemer-IN.SG' referring to a Turkish location was corrected to *Keeris* 'vortex.NOM.SG'. Such behaviour was the most characteristic to Vabamorf which also proposed changes to rather common first names, e.g., *Nadja∼Andja* 'giver'. Furthermore, some unnecessary edits made by Google, Word and Symspell were caused by splitting compound words.

On the other hand, it should be noted that the statistical spell-checkers do not correct capitalization because all words are transformed to lowercase when processing the text and then printed in the original casing. In general, all the tested spelling correction tools struggled with proposing the right correction instead of a candidate with a smaller edit distance (e.g., **musiika∼muusika* 'music' was corrected as *mustika* 'blueberry.GEN.SG'; **sõidata∼sõita* 'ride-INF' as *sõimata* 'curse-INF').

In conclusion, two of the tested statistical spell-checkers achieved a better precision and recall in correcting Estonian L2 learners' spelling errors compared to the existing open-source speller Vabamorf . Jamspell's performance was similar to MS Word in error detection and comparable with Google in error correction, the main difference being that Google corrected more spelling errors at the cost of making more unnecessary edits. Therefore, Jamspell should be favoured if the priority is to minimize needless corrections.

4 Jamspell Correction Models

We experimented with different training data to see if we can improve Jamspell's efficiency in learner spelling error detection and correction. The training sets are listed in table 5.

Training corpus	Sentences	Words
Web 2019	40,880,346	512,567,596
Reference + Wikipedia + DOAJ	16,935,524	230,066,343
Reference	13,173,122	180,944,778
Web 2019 sample	6,000,000	75,237,791
Reference sample	6,000,000	82,401,187
Reference + Web 1:1	6,000,000	78,855,570
Reference + Web 10:1	6,600,000	89,921,477
Reference + Wikipedia + DOAJ sample	4,172,777	55,743,160

Table 5: Data for training Jamspell models.

On the one hand, we combined the Estonian Reference Corpus with the DOAJ and Wikipedia corpora of ENC 2019. These subcorpora contain, to a large extent, language-edited texts. As the Reference Corpus constitutes the majority of this

training set, we also extracted a more balanced sample, which includes an equal amount of randomly chosen sentences from the Reference Corpus and Wikipedia corpora as well as the whole DOAJ corpus (442,663 sentences). On the other hand, we trained Jamspell on the Estonian Web corpus 2019 that comprises a more diverse selection of texts, from informal blog posts and forum discussions to periodicals and educational materials. We used the full corpus and a sample similar in size with the Reference Corpus sample. Thirdly, we merged the Reference Corpus and Web 2019 material in an equal ratio and in a ratio of 10:1, giving emphasis to the more "standardized" texts and using the web texts to add variation to the dataset.

The results of spelling error detection and correction obtained on the previously used test set are presented in tables 6 and 7. The system edit distribution is provided in table 8. In case of similar training datasets (full corpus and sample), the lower-performing correction model has been omitted. Models trained on samples of the Reference Corpus and its combination with other edited subcorpora achieved better or similar results compared to the models trained on full text sets. Contrary to that, the model trained on the whole Estonian Web 2019 performed better than the model based on the web sample in all aspects.

The comparison of the Jamspell models reflects the well-known trade-off between precision and recall. The highest error detection and correction precision were achieved by the model trained on Estonian Web 2019. It was the least likely to make unnecessary corrections but also to detect words with a spelling error, thus having the lowest recall. At the same time, the initial model trained on a Reference Corpus sample scored highest in error detection and correction recall, being able to identify and correct the largest amount of spelling errors. The latter model featured the best F0.5 score in error detection, whereas the Web 2019 model had a slightly better F0.5 score in error correction.

In terms of spelling error detection, the 10:1 Reference + Web sample offered a compromise, yielding a higher precision than the Reference Corpus model and a higher recall than the Estonian Web model. This resulted in the second best F0.5 score. On the other hand, there was little variation in the error correction F0.5 score. The performance obtained with the 10:1 Reference + Web sample was almost identical to the model trained on the Reference + Wikipedia + DOAJ sample. The 1:1 Reference + Web sample model scored slightly higher in correction precision and lower in correction recall.

Training corpus	F0.5	Precision	Recall
Reference sample	**83.9**	89.6	**67.0**
Reference + Web 10:1	82.7	91.2	60.2
Web 2019	81.9	**94.3**	53.7
Reference + Wikipedia + DOAJ sample	80.4	87.7	60.2
Reference + Web 1:1	79.9	89.6	55.7

Table 6: Spelling error detection metrics of Jamspell models (%), ranked by F0.5 score.

Training corpus	F0.5	Precision	Recall
Web 2019	**64.7**	**74.4**	42.4
Reference sample	64.1	68.4	**51.1**
Reference + Wikipedia + DOAJ sample	63.5	69.3	47.6
Reference + Web 10:1	63.1	69.6	46.0
Reference + Web 1:1	63.1	70.8	44.0

Table 7: Spelling error correction metrics of Jamspell models (%), ranked by F0.5 score.

Concerning the relation between the training corpus type and size, and the performance of the spell-checking model, we may infer that a smaller, "standard language" dataset rather facilitates higher recall. Increasing the dataset introduces more noise, thus the errors are outlined less clearly. A much larger and more diverse language model leads to higher precision; decreasing the dataset reduces lexical variation and entails more unnecessary edits. For comparison, the Web 2019 trigram model consists of 279.1 million trigrams, whereas the model trained on the Reference Corpus sample has 52.8 million trigrams.

The choice of the most suitable model depends

Training corpus	Errors detected	Full corrections	Partial corrections	Broken words
Reference sample	**207**	**129**	**29**	24
Reference + Wikipedia + DOAJ sample	186	122	25	26
Reference + Web 10:1	186	116	26	18
Reference + Web 1:1	172	113	23	20
Web 2019	166	115	16	**10**

Table 8: Jamspell models ranked by spelling errors detected and corrected.

on the purpose – whether we want to maximize the amount of errors detected and corrected, minimize the amount of needless corrections, or find a middle ground. For this, combining a larger proportion of standard texts with a smaller proportion of web material seems the best suited. In summary, the results are promising compared to the precision and recall of learner spelling error correction accomplished in other languages (e.g., Bexte et al. 2022; Kantor et al. 2019).

Three best-performing Jamspell models have been made available for use as a part of the new Estonian spelling and grammatical error correction toolkit currently in development[8].

5 Conclusion and Perspectives

This study has demonstrated the benefit of statistical context-sensitive spelling correction for processing L2 learner writings. Jamspell that uses trigram contexts of words for spell-checking could correct real-word errors and other learner-specific spelling errors more efficiently than other tested open-source spellers. In spelling error correction, it also outperformed MS Word speller, achieving precision and recall comparable to Google's corrector. In spelling error detection, its performance was similar to MS Word's and better than Google's. The evaluation of different Jamspell correction models revealed that using a web corpus as training material increases error detection and correction precision, while using a reference corpus increases recall.

We consider the current correction models a decent baseline for further development. Their performance could be improved, e.g., by employing learner spelling error frequency data or named-entity recognition to avoid false name edits and enable correction of name capitalization.

We acknowledge that the results might have been different if we had implemented Norvig's spell-checking algorithm on a trigram language model. The tested spell-checking tools and models should also be evaluated on a larger error-annotated set of writings by L2 learners as well as native speakers. Such a gold-standard dataset of approximately 8,000 sentences is in development for Estonian. Expectedly, context-sensitive spelling correction also benefits proficient language users, although the difference in performance may not be as outstanding.

[8]The repository of the collaborative project with the University of Tartu can be accessed at `https://koodivaramu.eesti.ee/tartunlp/corrector/-/tree/main/`.

Acknowledgments

This research has been funded by the national programme "Estonian Language Technology 2018-2027" and the Tallinn University Research Fund. We thank Marko Kollo, Rico-Andreas Lepp and Martin Mõtus for their help in testing the spell-checking tools. We also thank Kaisa Norak, Linda Luig and Pille Eslon for working on error annotation.

References

M. Bexte, R. Laarmann-Quante, An. Horbach, and T. Zesch. 2022. Lespell – a multi-lingual benchmark corpus of spelling errors to develop spellchecking methods for learner language. In *Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022)*, pages 697–706.

W. Garbe. 2017. https://towardsdatascience.com/sym spellcompound-10ec8f467c9b 1000x faster spelling correction. *Towards Data Science*.

T. Heift and A. Rimrott. 2008. Learner responses to corrective feedback for spelling errors in call. *System*, 36(2):196–213.

H.-J. Kaalep, F. Pirinen, and S. N. Moshagen. 2022. You can't suggest that?! comparisons and improve-

ments of speller error models. *Nordlyd*, 46(1):125–139.

Y. Kantor, Y. Katz, L. Choshen, E. Cohen-Karlik, N. Liberman, A. Toledo, A. Menczel, and N. Slonim. 2019. Learning to combine grammatical error corrections. In *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*, pages 139–148.

K. Koppel and J. Kallas. 2020. https://doi.org/10.15155/3-00-0000-0000-0000-08565L Estonian national corpus 2019. *Towards Data Science*.

S. Kumar and S. Tong. 2019. https://workspace.google.com/blog/ai-and-machine-learning/using-neural-machine-translation-to-correct-grammatical-in-google-docs Using neural machine translation to correct grammatical faux pas in google docs. *Google Workspace*.

J. Lawley. 2016. Spelling: computerised feedback for self-correction. *Computer Assisted Language Learning*, 29(5):868–880.

H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant. 2014. The conll-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14.

P. Norvig. 2007. https://norvig.com/spell-correct.html How to write a spelling corrector.

Author Index